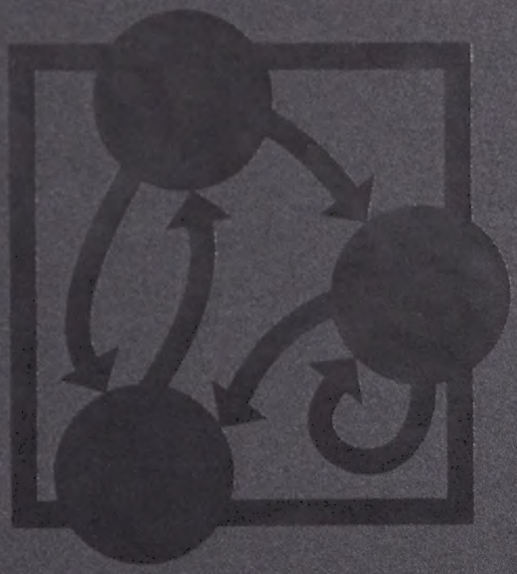
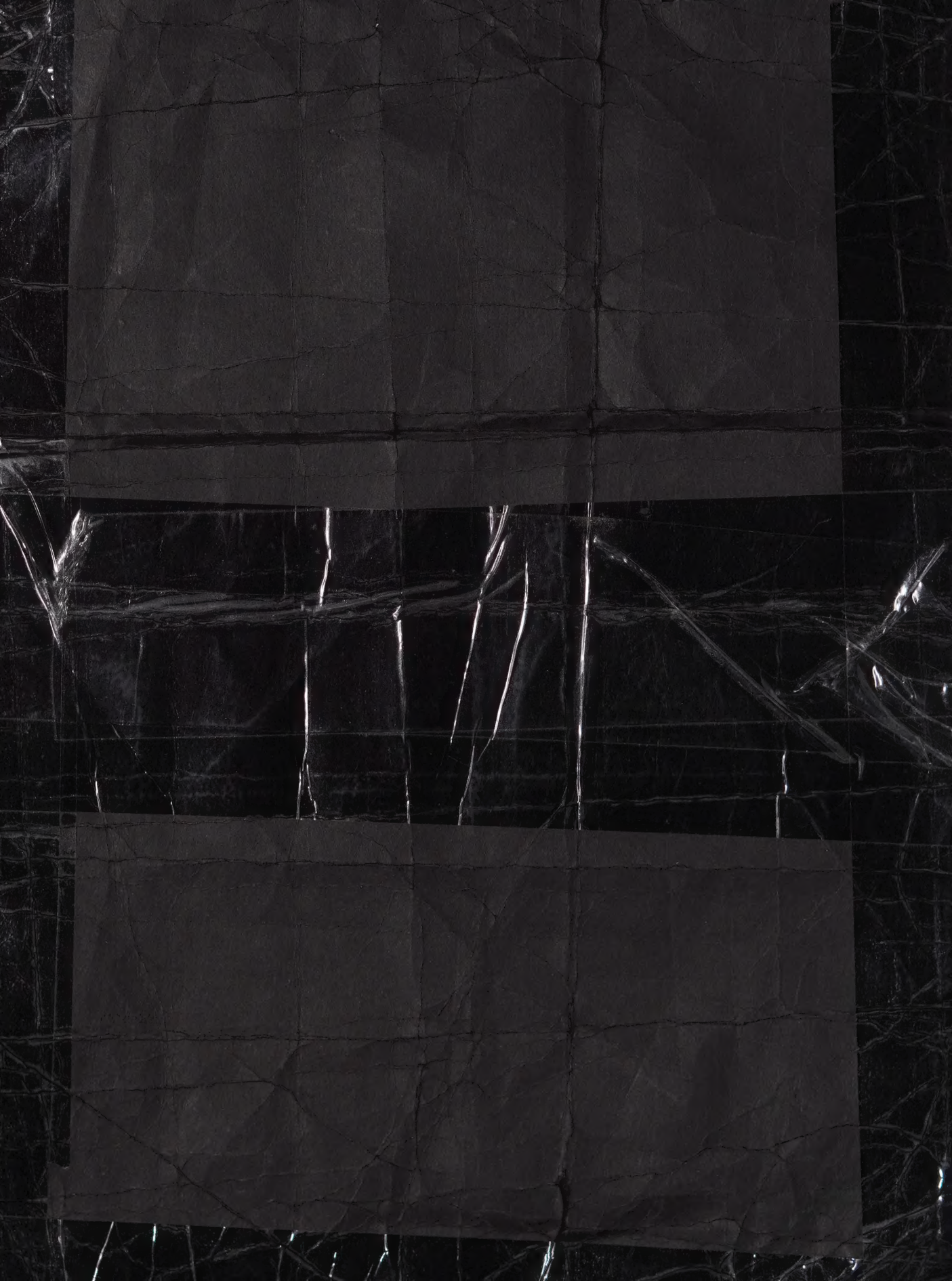
GENERAL SIGNAL

Networks™



INTERVIEW® 8000 SERIES

TECHNICAL MANUAL



Introduction

Congratulations on your choice of one of the INTERVIEW 8000 *TURBO* Series line of Data Protocol Analyzer/Emulators! This wise decision will be rewarded with many productive years of testing and emulation, along with the capability of upgrading your unit should your network require it.

For your ease-of-use, we have provided you with four (or five*) texts (and the Easy View On-Line Menu System):

30 Minutes to Programming the INTERVIEW® 8000 Series, 951-B0431-01
INTERVIEW® 8000 Series Technical Manual, September 1995, Volume I (951-B0424-01)
INTERVIEW® 8000 Series Technical Manual, September 1995, Volume II (951-B0427-01)
INTERVIEW® 8000 TURBO Series Quick Reference Guide to the C Library (951-B0430-01)
INTERVIEW® 8000 ATM Technical Manual, September 1995, (951-B0682-01) [with ATM units only]*

30 Minutes to Programming the INTERVIEW 8000 Series

Begin with the "thirty-minute guide." There is a data disk provided with it to introduce you to and to give you a crash course in general operation of the INTERVIEW 8000 *TURBO* units. You'll find this a half-hour well spent as you get a feel for using the function keys and menus, familiarizing yourself with the operation of the INTERVIEW.

INTERVIEW® 8000 Series Technical Manual

The two volumes of the *INTERVIEW® 8000 Series Technical Manual* are divided into *Part I: Basic Operation* and *Part II: Advanced Programming*. These are the complete source of information on your unit and so are understandably quite large. We do *not* expect you to read them in their entirety; therefore, the Table of Contents and the *three* comprehensive indexes—all three are multi-level and cross-referenced—are valuable tools to use in familiarizing yourself with the technical manual. There is also a List of Tables following the Table of Contents in the front matter of the first volume.

Part I: Basic Operation. A glance at the Table of Contents reveals the first volume consists of 49 sections grouped into twelve sub-headings, including an index for the first volume:

I	Overview
II	Setup and Display
III	BCC and BERT
IV	Interfaces
V	File System and Miscellaneous Utilities
VI	Statistics
VII	Programming Concepts: Tiers and Layers
VIII	Trigger Menus
IX	Protocol Spreadsheet
X	OSI Primitives
XI	Protocol Packages
XII	Index

Part I defines the basic operation of the INTERVIEW in detail and describes the menu fields and their entries in depth.

Part II: Advanced Programming. The Table of Contents shows the second volume consists of Sections 50 through 83. These sections are grouped under four sub-headings, along with Appendixes A through J and two indexes—one combined index for both volumes of the technical manual and the second index for the C-language structures, variables, and routines defined in the second volume:

XIII Theory of Operation
XIV C Language
XV C Library
XVI Protocol Library
Appendixes
Index

Part II is technically for the advanced C-language programmer who wishes to modify programs already provided in the software or to create his own programs. Along with a theory of operation of the unit and C-language background, it provides the C-language and protocol structures, variables, and routines used in the INTERVIEW 8000 *TURBO* Series. Structures are defined in depth; variables are discussed in detail with the type, values (hex and decimal), and meanings presented in a tabular format. The routines are particularly well documented with a synopsis, a description, inputs, returns, and examples given for each routine as applicable. The quick reference guide (described below) is a companion to this second volume of the technical manual.

Quick Reference Guide to the C Library

The quick reference guide is a condensed version of the C-language structure, variable, event, and routine information used in the INTERVIEW as defined in the second volume of the technical manual. This small reference guide is easy to use and cross-references pages in the technical manual where the C-language terms are defined and used.

It is divided into three parts: an alphabetic listing, a listing by function, and an alphabetic definitions listing. If you do any programming on the Protocol Spreadsheet with the INTERVIEW, you will find this handy little guide to be very useful.

Easy View On-Line Menu System

When you boot up the INTERVIEW, the Easy View Main Menu is the first menu displayed. You'll find this on-line documentation to be a valuable tool. Operational information, help, tutorials, and installed application programs are available at your screen in this menu system at the touch of a key. Follow the instructions given on the bottom of the display to access the desired screen menus. Detailed information on using Easy View is documented in the technical manual in Sections 4, 19, and 20.

ATM (Asynchronous Transfer Mode) Technical Manual and Associated Manuals

If your unit is an INTERVIEW 8800 *PLUS* ATM or INTERVIEW 8750 ATM *EXPRESS*, you will also be receiving the ATM technical manual. This manual describes the ATM units and is specific to Asynchronous Transfer Mode testing. Included with it are various manuals for the on–line ATM application programs.

Again, congratulations on your selection of an INTERVIEW 8000 *TURBO* Series unit...we hope these manuals will serve to answer your questions before you need to ask them.

INTERVIEW® 8000 *TURBO* Series
Technical Manual

Issue 3, September 1995

GENERAL SIGNAL
Networks

13000 Midlantic Drive
Mount Laurel, New Jersey 08054
Telephone: (609) 234-7900
FAX: (609) 778-8700
Internet: http://www.telenex.com

© 1992,1995 General Signal Networks

Notice

This is the July 1997 addendum (951-B0823-01) to Issue 3 of *INTERVIEW® 8000 TURBO Series Technical Manual,* (951-B0424-01 and 951-B0427-01), September 1995.

This addendum was prompted by the release of Operating System Software Revision 14.02. These revisions:

- Addresses changes made to the Remote Setup Menu to facilitate Ethernet connectivity, adding an NTP server address field and a field to identify your time zone difference from Coordinated Universal Time (UTC).
- Adds Year 2000 compliance to the INTERVIEW by specifying the year as a 4−character field. This fix also adds second resolution to the time and permits a Year 2000 leap year adjustment to be made automatically.

For Volume I of the manual (951-B0424-01):

At this page in the text:	*Do this:*
Page iii/iv	Add this page *on top of* the existing Page(s) iii/iv for reference.
Pages 2-15 through 2-20	Replace old pages 2-15 through 2-20 with new pages 2-15 through 2-20. Addition of the new Remote Setup Screen fields to select an NTP server for time synchronization and to reflect the UTC offset for the time zone in which the INTERVIEW is operating.
Pages 17-1 through 17−4	Replace old pages 17−1 through 17−4 with new pages 17-1 through 17−4. Modifies the date field for a 4−digit year instead of a 2−digit year (Year 2000 compliance); also changes the *Set Time* field to allow you to designate time changes as HH:MM:SS.

For Volume II of the manual (951-B0427-01):

At this page in the text:	*Do this:*
Pages 66-1 through 66-2	Replace old pages 66-1 through 66-2 with new pages 66-1 through 66−2. Changes the text to more accurately reflect the INTERVIEW Remote Control options available.

Notice

This is the December 1996 addendum (951-B0804-01) to Issue 3 of *INTERVIEW® 8000 TURBO Series Technical Manual,* (951-B0424-01 and 951-B0427-01), September 1995.

This addendum was prompted by the release of Operating System Software Revision 14.00 and it's concurrent firmware revision 10.00. These revisions:

- Allows you to choose a 32−bit file system instead of the previous standard 16−bit file system. This removes the file system size restriction of 32 Mbytes.
- Changes references to specific SCSI hard disk drive sizes and reflects the current standard disk size of 2.1 Gbytes when appropriate.
- Adds information on Ethernet connectivity, including position and type of connector. Information on the Remote Setup Menu was also added to the manual.

For Volume I of the manual (951-B0424-01):

At this page in the text:	*Do this:*
Page iii/iv	Add this page *on top of* the existing Page(s) iii/iv for reference.
Pages v through xx	Replace old pages v through xx with new pages v through xxii. This updates the Table of Contents to reflect added/deleted material.
Pages xxi through xxiv	Replace old pages xxi through xxiv with new pages xxiii through xxviii. This updates the List of Tables to reflect added/deleted material.
Pages 1-1 through 1-20	Replace old pages 1-1 through 1-20 with new pages 1-1 through 1-20. Addition of the new 10BASE−T (RJ−45) Ethernet connector, which replaces the RS−170 video connector for newer INTERVIEW units.
Pages 2-1 through 2−30	Replace old pages 2-1 through 2-30 with new pages 2-1 through 2-34. Addition of Remote Control Setup menu descriptions and instructions on how to customizing Modem Script selections; also changes reflecting a *generic* SCSI hard disk drive instead of specific sizes.
Pages 7-1 through 7−12	Replace old pages 7-1 through 7-12 with new pages 7-1 through 7-12. Addition of information pertaining to the new 32−bit file system and DAT sizes for 2.1−Gbyte hard disk drives; also changes reflecting a *generic* SCSI hard disk drive.
Pages 7-1 through 7−12	Replace old pages 7-1 through 7-12 with new pages 7-1 through 7-12. Addition of information pertaining to the DAT sizes for 2.1−Gbyte hard disk drives; also changes reflecting a *generic* SCSI hard disk drive.
Page 12-1/12-2	Replace old page 12-1/12-2 with new page 12-1/12-2. Deletion of table of standard TIMs with specific INTERVIEW units and revision of text to reflect a generic presentation for TIMs.

Pages 14-1 through 14-18	Replace old pages 14-1 through 14-18 with new pages 14-1 through 14-20. Information was added concerning the 2.1−Gbyte SCSI hard disk drive and the 32−bit file system option for formatting hard disk drives.
Pages 15-1 through 15-24	Replace old pages 15-1 through 15-24 with new pages 15-1 through 15-28. Information on the number of files and/or directories supported by the 32−bit file system and on the use of wildcards in file and directory names.
Pages 49-13 through 49-16	Replace old pages 49-13 through 49-16 with new pages 49-13 through 49-16. Corrects artwork for the ATM cell structure to correct a typographical error.
Pages Index A-1 through A-32	Replace old pages Index A-1 through Index A-32 with new pages Index A-1 through Index A-32. Updates the Volume I Index to reflect the changes made in this addendum.

For Volume II of the manual (951-B0427-01):

At this page in the text:	*Do this:*
Pages 81-1 through 81-12	Replace old pages 81-1 through 81-12 with new pages 81-1 through 81-12. Changes the ISDN−specific structures and routines to reflect the addition of dual B−channel access functionality. Also addresses changes to the EOC routines for the U interface.
Pages E-37 through E-40	Replace old pages E-37 through E-40 (Appendix) with new pages E-37 through E-40. Corrects a typographical error in the pinout tables for the EIA−530 connector.
Pages Index B-1 through B-68	Replace old pages Index B-1 through Index B-68 with new pages Index B-1 through Index B-70. Updates the Volume I and Volume II Combined Index to reflect the changes made in this addendum.

Notice

This is the November 1995 addendum (951-B0745-01) to Issue 3 of *INTERVIEW® 8000 TURBO Series Technical Manual,* (951-B0424-01 and 951-B0427-01), September 1995.

This addendum was prompted by the release of Operating System Software Revision 12.00 and the merger of Telenex, Data Switch, and Tau–Tron into the newly formed company of General Signal Networks. This revision introduces the use of wildcard symbols in File Maintenance operations, adds information on two TIM Expansion Shelf routines and on new operator messages, and clarifies a statement on mounting the equipment in a rack.

At this page in the text:	*Do this:*
Title page	Replace the old title page with the new title page. Company logo change.
Notice page	Replace the old notice page with the new notice page. Company name change from Telenex Corporation to "General Signal Networks – Mount Laurel" and addition of Internet access information.
Pages 15-7 through 15-24	Replace old pages 15-7 through 15-24 with new pages 15-7 through 15-26. Addition of wildcard symbols into File Maintenance operations and new COPY command information.
Page 81-17/81-18	Replace old page 81-17/81-18 with new pages 81-17 through 81-20. Addition of two new TIM Expansion Shelf routines, *get_xtim_config* and *xtim_load_program*.
Pages A1-1 through A1-22	Replace old pages A1-1 through A1-22 with new pages A1-1 through A1-24. Additional error messages displayed for File Maintanance operations.
Pages G-1 through G-4	Replace old pages G-1 through G-4 with new pages G-1 through G-4. Company name change.
Page I-3/I-4	Replace old page I-3/I-4 with new page I-3/I-4. Addition of a statement regarding Telenex Corporation's recommendation of installing a writing shelf with the rack mount for the INTERVIEW "for proper installation for ergonomic keyboard use."

Notice

This is Issue 3 of the *INTERVIEW® 8000 Series Technical Manual*, September 1995, Volume I (951−B0424−01) and Volume II (951−B0427−01).

Issue 3 with its addendum (November 1995) is written to specifications for the INTERVIEW 8000 Series—the 8100 *TURBO*, the 8200 *TURBO*, 8600 *TURBO*, 8700 *TURBO*, 8800 *TURBO*, and their respective specialized units—with the Errata providing for software revision 12.00 or greater. Refer to the Hardware section of this manual for specific model descriptions; for ATM units, refer to the Hardware section of the *INTERVIEW® 8000 Series ATM Technical Manual, Asynchronous Transfer Mode (Broadband),* (951−B0424−01). In most instances, further software revisions will be accompanied by an addendum to this issue. In cases where new software does not affect the accuracy of the manual, however, an addendum may not be produced.

This issue of the manual (with addendum) adds the following features to the INTERVIEW:

- Hardware information added for the following:
  - new power supply voltage selection
  - change of fuse from 4 amp to 5 amp
  - new internal cables
  - MPM board changes
- New information added to the Setup Menu screen *when an ATM Interface Module is installed* in an ATM unit:
  - a new ATM Interface Module function key appears
  - AIM Interface Setup screen and with DISABLE function accessible from softkey rack
  - use of and results of using the new DISABLE function key
- Addition of AAL1 information in ATM section.
- Addition of wildcard symbols into File Maintenance operations and new COPY command information.
- Addition of four new user routines for High-Speed Frame Mode, SMDS, and ATM libraries: *frame_disable_dce_1*, *frame_disable_dte_1*, *frame_disable_dce_2*, and *frame_disable_dte_2*.
- Addition of two TIM Expansion Shelf routines, *get_xtim_config* and *xtim_load_program*.
- Addition of new error messages displayed for File Maintenance operations.
- "Communications with Telenex" appendix updated with new information.

General Signal Networks is a newly formed company by the merger of Telenex, Data Switch, and Tau−Tron. Telenex Corporation is now General Signal Networks − Mount Laurel; we retain the same personnel and high standards which have always been hallmarks of Telenex Corporation and AR Test Systems. *Any references within this manual to AR, AR Test Systems, and Telenex Corporation now refer to General Signal Networks − Mount Laurel.*

General Signal Networks − Mount Laurel reserves the right to improve this manual or the equipment it describes without prior notice. Any references to upgraded equipment, options,

accessories, and software packages made in the manuals may not apply to your present unit. Contact Customer Service for information on such upgrades or other purchases.

Any duplication of the material in this manual in any form without written permission from General Signal Networks – Mount Laurel is strictly forbidden.

General Signal Networks – Mount Laurel will not be held responsible for any damages incurred in the use of this product.

For technical information, programming assistance, error decoding, and repairs, contact the factory. General Signal Networks – Mount Laurel is located in Mount Laurel, New Jersey, approximately 15 miles east of Philadelphia, PA. Local customers should call (609) 234-7900; in the northeastern United States, call (800) 222-5482; and in the rest of the U.S., call (800) 222-0187. Access us on the Internet at *http://www.telenex.com*.

Address questions and comments about this manual and other technical publications to the Technical Writing Department on extension 3548 at these same telephone numbers.

Contents

Introduction

Part I: Basic Operation

I Overview

1 Hardware 1–1

1.1 Physical Dimensions 1–3
1.2 Keyboard 1–3
1.3 Front Panel 1–3
1.4 Disk Drives 1–5
1.5 Back Panel 1–8
1.6 Storage Capacity 1–15
1.7 Clock 1–16
1.8 Operating Environment 1–16
1.9 Operating Positions 1–17
1.10 Power Up 1–18

2 General Operation 2–1

2.1 Power Up 2–1
2.2 Initializing System Software (Booting Up) 2–3
2.3 Installing New System Software on Hard Disk 2–7
2.4 Backing Up the Hard Disk 2–8
2.5 The Menus 2–9
2.6 Customizing the Modem Script Field Selections 2–19
2.6 Running a Test Program 2–20
2.7 Data Flow 2–24
2.8 How to Correct Common Problems 2–26
2.9 Optimizing Recording Rates 2–31
2.10 Optimizing Analysis 2–34

3 Keyboard 3–1

3.1 Hard Keys and Softkeys 3–3
3.2 Programming Keys 3–3
3.3 Real–Time Keys 3–10
3.4 Freeze–Mode Keys 3–13
3.5 Remote Control Keys 3–14

4 Easy View 4–1

4.1 Booting Up 4–3
4.2 Entering and Exiting Easy View 4–4
4.3 Easy View Menus 4–4
4.4 Using Easy View 4–8
4.5 Additional Easy View Functions 4–17
4.6 Installing Easy View Updates 4–19

II Setup and Display

5 Line Setup 5–1

5.1 Mode 5–3
5.2 Source 5–5
5.3 Code 5–6
5.4 Bits 5–7
5.5 Parity 5–7
5.6 Format 5–7
5.7 Clock Source 5–15
5.8 Bit–Order/Polarity 5–16
5.9 NRZI 5–17
5.10 MIL 5–17
5.11 Sample Line Setups 5–17

6 Run–Mode Display 6–1

6.1 Auto Configure 6–3
6.2 Entering Run Mode 6–5
6.3 Selecting Character–Data Display 6–7
6.4 Special Features of Data Display 6–13
6.5 Protocol Trace 6–17
6.6 Program Trace 6–19
6.7 Statistics Display 6–22
6.8 Display Window 6–23
6.9 User Trace 6–25
6.10 Display Correlation in Freeze Mode 6–27
6.11 No Display 6–28

7 Record Setup 7–1

7.1 Format of Recorded Data 7–3
7.2 Recording Medium 7–3

7.3 The Screen Buffer 7–4
7.4 SCSI Drive Maximum Recording Times 7–5
7.5 The Record Setup Screen 7–5
7.6 Trigger Control of Capture 7–11
7.7 Spreadsheet Control of Recording 7–11
7.8 Manual Control of Recording 7–12

8 Layer Setup 8–1

8.1 Personality Packages 8–3
8.2 Selecting and Loading Protocols 8–3
8.3 The Protocol Configuration Screen 8–4
8.4 Saving the Layer Setup Screen 8–5

9 FEB Setup 9–1

9.1 Buffering Idle, Control Leads, Ticks, and Timestamps 9–3

III BCC and BERT

10 Block Checking 10–1

10.1 BCC Symbols 10–3
10.2 BCC Conditions 10–5
10.3 Transmitted BCC 10–5
10.4 Standard BCC Parameters 10–6
10.5 BCC Setup Menu Fields 10–11
10.6 BISYNC vs. Selectable CRC Mode 10–14

11 Bit Error Rate Testing 11–1

11.1 Pseudorandom Bit Patterns 11–3
11.2 Test Configurations 11–4
11.3 BERT Operation: Full Duplex 11–4
11.4 BERT Operation: Half Duplex 11–6
11.5 BERT Setup Screen 11–7
11.6 Transmission Format: Line Setup Menu 11–14
11.7 Run Mode: Keyboard Control 11–17
11.8 Run Mode: Status Line 11–18
11.9 Run Mode: Statistical Display 11–19
11.10 Loopback at the Transmitting INTERVIEW 11–21

IV Interfaces

12 Standard Interfaces 12–1

13 Optional Interfaces 13–1

V File System and Miscellaneous Utilities

14 Disk Maintenance 14–1

14.1 The Disks 14–3
14.2 Allocating Disk Space 14–3
14.3 Data Acquisition Tracks vs. the Filing System 14–3
14.4 The Disk Maintenance Screen 14–4

15 File Management 15–1

15.1 Loading and Saving Files 15–3
15.2 Orienting Yourself in the Filing System 15–4
15.3 The File Maintenance Screen 15–11

16 Printer Control 16–1

16.1 The Connector 16–3
16.2 Configuring the Printer Setup Screen 16–3
16.3 Saving the Printer Setup 16–7
16.4 How to Print Static Displays 16–8
16.5 How to Print Data 16–10
16.6 Spreadsheet Control of Printing 16–17
16.7 Printing Disk Files 16–17

17 The Time–of–Day Clock 17–1

18 Color Display 18–1

18.1 Connectors for External Monitors 18–3
18.2 Color Control from the Miscellaneous Utilities Screen 18–3
18.3 Black and White Data Enhancements 18–6

19 Easy View Setup 19–1

19.1 Enable Easy View 19–3
19.2 Enter Easy View After Power–Up 19–4

19.3 Keep Easy View Menu Information in Memory 19–4
19.4 Display Program Warning Messages 19–4
19.5 Load Program Edit Timeout 19–5
19.6 Screen Saver Timeout 19–5

20 Easy View Maintenance 20–1

20.1 The Easy View Menu File 20–3
20.2 Accessing the Maintenance Program 20–4
20.3 Automatic Installation 20–5
20.4 Customizing Programs 20–6
20.5 Script File Format 20–20
20.6 Creating a Text (Help) File 20–33
20.7 Easy View Directory Structure 20–34

VI Statistics

21 Tabular Statistics 21–1

21.1 Counters and Timers 21–3
21.2 Preparing the Tabular Statistics Screen 21–4
21.3 Sampling Current Values 21–6
21.4 Accumulators 21–7
21.5 Keeping a Statistical Log 21–9
21.6 The Sampling Action as Divisor 21–11

22 Graphic Statistics 22–1

22.1 Enabling the Graphic Display 22–3
22.2 Cursor Movement on Graphical Statistics Menu 22–4
22.3 Menu Fields 22–4

VII Programming Concepts: Tiers and Layers

23 Three–Tiered Programming 23–1

23.1 Trigger Setup Screens 23–2
23.2 The Protocol Spreadsheet 23–3
23.3 C Programming Language 23–4
23.4 Integrating Programming Methods 23–4

24 The Layered Program Model 24–1

24.1 States 24–3

24.2 Tests 24–3
24.3 Layers and the OSI Model 24–4
24.4 Personality Packages 24–7
24.5 Primitives 24–8
24.6 Constants 24–8

VIII Trigger Menus

25 Trigger Conditions 25–1

25.1 Active Triggers 25–4
25.2 Combining Conditions on the Same Trigger Setup Screen 25–4
25.3 Receiver 25–5
25.4 EIA 25–9
25.5 Timeout 25–9
25.6 Transmission Complete 25–10
25.7 Internal Flag Bits 25–10
25.8 Buffer Full 25–11
25.9 Counter 25–11
25.10 Keyboard 25–12

26 Trigger Actions 26–1

26.1 Displaying a Prompt 26–3
26.2 Transmitting 26–4
26.3 Internal Flags 26–6
26.4 Enhancing the Display 26–7
26.5 Controlling Timeouts 26–8
26.6 Counters 26–8
26.7 Timers 26–10
26.8 Alarm 26–11
26.9 Capture of Data in the Screen Buffer 26–11

27 The Trigger Summary Screen 27–1

IX Protocol Spreadsheet

28 Programming Blocks 28–1

28.1 Before You Begin a Spreadsheet Program 28–3
28.2 Creating a Spreadsheet Program 28–3
28.3 Program Structure 28–6
28.4 Compiled Spreadsheet 28–10

28.5 Configuring the Size/Number of IL Buffers 28–13
28.6 Comments in a Spreadsheet Program 28–14

29 Constants 29–1

29.1 Definition of Constants 29–3
29.2 Constant Names 29–4
29.3 Scope 29–4
29.4 Referencing Constants 29–5
29.5 Nested Constants 29–6
29.6 Precedence 29–6
29.7 Expansion 29–7

30 Editor 30–1

30.1 Basic Editing Functions 30–3
30.2 Editing Function Keys 30–5

31 Layer–Independent Conditions and Actions 31–1

31.1 Naming Requirements 31–1
31.2 Rules for Combining Conditions 31–2
31.3 Layer–Independent Conditions 31–3
31.4 Layer–Independent Actions 31–8

32 Layer 1 Conditions and Actions 32–1

32.1 Single–Port Layer 1 Conditions 32–1
32.2 Dual–Port Layer 1 Conditions 32–6
32.3 Single–Port Layer 1 Actions 32–8
32.4 Dual–Port Layer 1 Actions 32–17

33 Strings 33–1

33.1 Strings To Be Matched Against Line Data 33–3
33.2 Strings To Be Transmitted 33–3

X OSI Primitives

34 OSI Primitives on the Protocol Spreadsheet 34–1

34.1 Softkey Selections 34–4
34.2 Sample Primitives: CONNECT INDs and CONNECT REQs 34–7
34.3 Sample Primitives: DATA INDs and DATA REQs 34–9

35 Automatic OSI Primitives 35–1

XI Protocol Packages

36 X.21 Layer 1 36–1

36 X.21 Layer 1 36–3

36.1 X.21 bis 36–4
36.2 Transmitter/Receiver Phases 36–4
36.3 Sending From Layer 2 36–5
36.4 X.21 Conditions 36–5
36.5 X.21 Actions 36–7

37 X.25 Layer 2 37–1

37.1 Frame–Level Setup 37–3
37.2 Protocol Trace 37–5
37.3 Monitor Conditions 37–11
37.4 Emulate–Mode Conditions 37–15
37.5 Emulate Actions 37–21
37.6 Display Actions 37–32
37.7 Automatic Primitives 37–33
37.8 Programming Example: Converting Protocol Bytes to Hexadecimal 37–35
37.9 Programming Example: A Simple "Automatic" Layer 2 X.25 Test 37–36

38 X.25 Layer 3 38–1

38.1 Packet–Level Setup 38–3
38.2 Protocol Trace 38–8
38.3 Monitor Conditions 38–15
38.4 Emulate–Mode Conditions 38–24
38.5 Emulate Actions 38–29
38.6 Display Actions 38–42
38.7 Automatic Primitives 38–44
38.8 Programming Example: Forcing Data Packets Out on the Line 38–45

39 SDLC 39–1

39.1 Frame–Level Setup 39–3
39.2 Protocol Trace 39–6
39.3 Monitor Conditions 39–12
39.4 Emulate–Mode Conditions 39–17

39.5 Emulate Actions 39–22
39.6 Display Actions 39–32
39.7 Automatic Primitives 39–34

40 SNA 40–1

40.1 Frame–Level Setup 40–3
40.2 SDLC Conditions and Actions 40–4
40.3 Protocol Trace 40–4

41 DDCMP Layer 1 41–1

41.1 Outsync 41–1
41.2 Block Checking 41–1

42 ISDN D Channel 42–1

43 LAPD 43–1

43.1 Frame–Level Setup 43–3
43.2 Protocol Trace 43–5
43.3 Monitor Conditions 43–9
43.4 Emulate–Mode Conditions 43–14
43.5 Emulate Actions 43–19
43.6 Display Actions 43–29
43.7 Automatic Primitives 43–31

44 Q.931 44–1

44.1 Protocol Trace 44–3
44.2 Monitor Conditions 44–9
44.3 Display Actions 44–12

45 SS#7 Layer 1 45–1

45.1 SS#7 Run–Time Displays 45–3
45.2 Setup for SS#7 45–3
45.3 SS#7 Compression at Layer 1 45–4
45.4 SS#7 Suppression at Layer 1 45–5

46 SS#7 Layer 2 46–1

46.1 Set Up for SS#7 46–3
46.2 Protocol Trace 46–3

46.3 Monitor Conditions 46–7
46.4 Display Actions 46–10
46.5 SS#7 Emulation 46–11
46.6 SS#7 Frame Structures and Values 46–11

47 SS#7 Layer 3 47–1

47.1 Packet–Level Setup 47–3
47.2 Protocol Trace 47–4
47.3 Monitor Conditions 47–7
47.4 Display Actions 47–11
47.5 Structure of SS#7 Message Signal Units 47–12

48 SMDS 48–1

49 ATM 49–1

49.1 Recording ATM over T1/E1 Data with the INTERVIEW 49–1
49.2 Overview 49–1
49.3 AAL 1 Protocol 49–4
49.4 AAL 3/4 Protocol 49–8
49.5 AAL 5 Protocol 49–14
49.6 Operations and Maintenance (OAM) Cells 49–17
49.7 ATM Framing: HEC and PLCP 49–18

XII Index

Index A: Part I of the Technical Manual ..... Index A–1

Part II: Advanced Programming

XIII Theory of Operation

50 Data Flow 50–1

50.1 Two Types of CPU 50–3
50.2 Front–End Buffer 50–5

51 Program Main 51–1

51.1 Translating a Simple Test into C 51–1
51.2 A Minimum of One Softkey State 51–2
51.3 Writing the Test Entirely in C 51–3

52 Regions in Spreadsheet 52–1

52.1 Layer and Test 52–1
52.2 State, Enter State, and Next State 52–2
52.3 Conditions and Actions 52–7
52.4 Example of Complete C Program 52–13
52.5 Summary of C Regions 52–14

53 Events 53–1

53.1 Example of Event: fevar_time_of_day 53–1
53.2 Various Origins of waitfor Events 53–2
53.3 Programming Considerations 53–3

54 Receiving and Transmitting Data 54–1

54.1 Locating Data in an IL Buffer 54–1
54.2 Monitor Path vs. Receive Path 54–3
54.3 Passing a Buffer Upwards 54–3
54.4 Layer 1 Transmit 54–4
54.5 Passing a Buffer Between Tasks 54–7
54.6 Sample Transmit Program: Sync or Async Echo 54–9
54.7 Sample Transmit Program: BOP Echo 54–10

XIV C Language

55 C Basics 55–1

55.1 Notable Variations in C 55–3
55.2 Editing a C Program 55–4

55.3 Error Reporting in C 55–4
55.4 Preprocessor Directives 55–5
55.5 Data Types 55–13
55.6 Operator Precedence 55–17
55.7 Strings 55–19
55.8 Recommended Sources 55–22

56 Variables 56–1
56.1 Creating or Accessing C Variables 56–1
56.2 Declaring Variables 56–2
56.3 Comparing a Variable to a Value 56–3
56.4 Checking a Variable in a waitfor Clause 56–4
56.5 Checking and Displaying Equivalent Values of a Variable 56–7
56.6 Isolating Bits from a Variable Value 56–8
56.7 Pointing to an Address 56–10
56.8 Creating a Character Pointer 56–11
56.9 Pointing with Subscripts 56–12
56.10 Creating a String 56–12
56.11 Comparing Strings 56–14
56.12 Accessing a Variable Inside a Structure 56–15
56.13 Creating a Structure Pointer 56–16

57 Routines 57–1
57.1 Declarations 57–1
57.2 Arguments 57–1
57.3 Returns 57–2
57.4 User–Defined Routines 57–3
57.5 Example Routines 57–4

XV C Library

58 Monitor/Transmit Line Data 58–1
58.1 Structures 58–1
58.2 Variables 58–2
58.3 Routines 58–7

59 EIA 59–1
59.1 Variables 59–1
59.2 Routines 59–3

60 Display Window and Trace 60–1
60.1 Current Display Mode 60–1
60.2 Softkey Display 60–3
60.3 Prompt Line 60–5

60.4 Display Window 60–5
60.5 Program and User Traces 60–31
60.6 Attributes 60–41
60.7 Protocol Trace Buffers 60–47

61 Counters, Timers, and Accumulators 61–1

61.1 Counters 61–1
61.2 Timers 61–8
61.3 Accumulators 61–12
61.4 Routines 61–14

62 OSI 62–1

62.1 Structures 62–3
62.2 Variables 62–13
62.3 Routines 62–37

63 Print 63–1

63.1 Structures 63–1
63.2 Variables 63–3
63.3 Routines 63–4

64 Disk I/O 64–1

64.1 Streams 64–1
64.2 Routines 64–3

65 Status 65–1

65.1 Unit Configuration 65–1
65.2 Current Display Mode 65–1

66 Remote Port I/O 66–1

66.1 Structures 66–1
66.2 Variables 66–2
66.3 Routines 66–4

67 AUX Port I/O 67–1

67.1 Variables 67–3
67.2 Routines 67–5

68 Other Library Tools 68–1

68.1 Structures 68–1
68.2 Variables 68–3
68.3 Routines 68–8

XVI Protocol Library

69 X.21 Library 69–1

69.1 Structures 69–1
69.2 Variables 69–2
69.3 Routines 69–4

70 X.25 Layer 2 Library 70–1

70.1 Structures 70–1
70.2 Variables 70–1
70.3 Routines 70–9

71 X.25 Layer 3 Library 71–1

71.1 Structures 71–1
71.2 Variables 71–1
71.3 Routines 71–10

72 SDLC Library 72–1

72.1 Structures 72–1
72.2 Variables 72–1
72.3 Routines 72–10

73 SNA Library 73–1

73.1 Structures 73–1
73.2 Variables 73–1
73.3 Routines 73–5

74 DDCMP Library 74–1

74.1 Structures 74–1
74.2 Variables 74–1
74.3 Routines 74–1

75 Secondary Channel Library 75–1

75.1 Structures 75–2
75.2 Variables 75–2
75.3 Routines 75–6

76 LAPD Library 76–1

76.1 Structures 76–1
76.2 Variables 76–1
76.3 Routines 76–9

77 Q.931 Library 77–1

77.1 Structures 77–1
77.2 Variables 77–1
77.3 Routines 77–5

78 SS#7 Layer 2 Library 78–1

78.1 Structures 78–1
78.2 Variables 78–1
78.3 Routines 78–5
78.4 SS#7 Layer 1 78–5

79 SS#7 Layer 3 Library 79–1

79.1 Structures 79–1
79.2 Variables 79–1
79.3 Routines 79–7

80 High–Speed Frame Mode Library 80–1

80.1 Sample Mode 80–1
80.2 Discarding Frames 80–1
80.3 Layer 1 Primitives for High–Speed Frame Capture Mode 80–2
80.4 Variables 80–2
80.5 Routines 80–3

81 TIM–Specific Library 81–1

81.1 Variables 81–1
81.2 ISDN S/T/U Structures 81–1
81.3 ISDN S/T/U Routines 81–2
81.4 ISDN Structures 81–15
81.5 ISDN Routines 81–15
81.6 Dual–Port T1 Structures 81–15
81.7 Dual–Port T1 Routines 81–15
81.8 Dual–Port G.703 Structures 81–15
81.9 Dual–Port G.703 Routines 81–15
81.10 Dual–Port DS–0A Structures 81–16
81.11 Dual–Port DS–0A Routines 81–16
81.12 Tim Expansion Shelf Structures 81–17
81.13 TIM Expansion Shelf Routines 81–18

82 SMDS Library 82–1

82.1 Structures 82–1
82.2 Variables 82–3
82.3 Routines 82–4

83 ATM Library 83–1
83.1 Structures 83–1
83.2 Variables 83–3
83.3 Routines 83–5

Appendixes

Appendix A: Operator Messages A–1
Appendix A1: Interactive Messages A1–1
Appendix A2: Easy View Messages A2–1
Appendix A3: Error Messages Issued by C Translator A3–1
Appendix A4: Error Messages Issued by C Compiler A4–1

Appendix B: Glossary of Acronyms, Abbreviations and Mnemonics B–1

Appendix C: Selectable Data Speeds C–1

Appendix D: Code Charts D–1
Appendix D1: Keyboard–to–Hex Translation D1–1
Appendix D2: Hex–to–Display Translation D2–1
Appendix D3: User–Defined Codes D3–1
Appendix D4: Keyboard–to–PC Conversion D4–1

Appendix E: Interface Specifications E–1

Appendix F: Packing and Shipping Instructions F–1

Appendix G: Communications with Telenex G–1

Appendix H: Field Service on the INTERVIEW 8000 Series H–1
Appendix H1: Eliminating Static Electricity H1–1
Appendix H2: Removing Logic Boards H2–1
Appendix H3: Installing Logic Boards H3–1
Appendix H4: Logic Board Components H4–1
Appendix H5: Enabling Automatic Reboot H5–1

Appendix I: Rack Mount (OPT–951–98–1) I–1

Appendix J: C Language Summary J–1

Index

Index B: Technical Manual Part I and Part II ..... Index B–1

Index C: C Structures, Variables, and Routines ..... Index C–1

List of Tables

Table 2-1	Summary Guide of Level1/Level 2 Timers and Maximum Number of Retransmissions for Specific Conditions	2–19
Table 2-2	Fields That Can Be Changed Without Causing Recompile	2–22
Table 3-1	Frequently Used Programming Keys	3–4
Table 5-1	Synchronization Parameters for Standard Codes	5–9
Table 6-1	Special Display Symbols	6–14
Table 7-1	SCSI Drive Recording	7–5
Table 7-2	RAM Size (bytes) for Normal and High–Speed Recording in the INTERVIEW 8000 TURBO Series Units	7–9
Table 10-1	Standard Block–Check Parameters for Sync or Async/Isoc Formats	10–7
Table 10-2	Block Check Polynomials	10–7
Table 10-3	BCC Setup Menu Fields	10–12
Table 11-1	Full– or Half–Duplex BERT (RS–232/V.24 Interface Control screen)	11–7
Table 11-2	BERT Pattern (BERT Setup screen)	11–8
Table 11-3	Half–Duplex BERT: Default Synchronization Points	11–13
Table 11-4	Sync or Async BERT (Line Setup screen)	11–17
Table 13-1	Optional Test Interface Modules	13–1
Table 15-1	Wildcards With File Designations	15–8
Table 15-2	File Types	15–14
Table 15-3	Source and Destination Copy Guidelines	15–21
Table 16-1	Character Representations on Menu Screen Printouts	16–7
Table 16-2	Character Representations on Data Printouts	16–10
Table 27-1	Abbreviations, Trigger Summary Conditions	27–4
Table 27-2	Abbreviations, Trigger Summary Actions	27–5
Table 33-1	Valid Entries in Receive Strings	33–2
Table 33-2	Valid Entries in Transmit Strings	33–6
Table 34-1	Primitive Prefixes and Associated Layers	34–4
Table 34-2	OSI Service Primitives	34–11
Table 37-1	Default Parameters in SEND Actions	37–23
Table 37-2	Automatic Primitives Generated at X.25 Layer 2	37–34
Table 38-1	Automatic Primitives Generated at X.25 Layer 3	38–44
Table 39-1	Automatic Primitives Generated at Layer 2 SDLC	39–35
Table 40-1	Fields in SNA Trace Display	40–8
Table 43-1	Automatic Primitives Generated in LAPD	43–31
Table 46-1	LSSU Status Field (Bits 2–0 of first Status Field Octet)	46–5

Table 47-1	MSU Service Indicators (Bits 3–0 of the SIO)	47–17
Table 47-2	Network Management Headers (Octet 9 for CCITT Format; Octet 12 for ANSI–US Format)	47–18
Table 47-3	SCCP Message Headers (Octet 9 for CCITT International or National Format; Octet 12 for ANSI–US Format)	47–19
Table 47-4	Telephone User Part (TUP) Message Headers	47–20
Table 47-5	ISUP Message Headers	47–22
Table 49-1	Pre–defined Header Field Values	49–13
Table 49-2	Payload Type Indicator Encoding	49–13
Table 55-1	Data Types: Ranges of Values Displayed and Printed	55–16
Table 55-2	Operator Precedence!	55–18
Table 55-3	C String Nonliterals	55–21
Table 58-1	Transmit Structures	58–1
Table 58-2	Monitor/Transmit Variables	58–3
Table 58-3	Character Buffer 16–Bit Word	58–18
Table 59-1	EIA Variables	59–2
Table 60-1	Current Display Variables	60–2
Table 60-2	Display Window Variables	60–6
Table 60-3	Display Window Buffer Structures	60–9
Table 60-4	Display Window/Trace Buffer 32–Bit Data Word	60–15
Table 60-5	Special Graphic Character Set	60–18
Table 60-6	Trace Buffer Structures	60–32
Table 60-7	Conversion Specifiers	60–46
Table 60-8	Protocol Trace Buffer Variables	60–48
Table 60-9	Protocol Trace Buffer Structures	60–50
Table 61-1	Counter Structures	61–2
Table 61-2	Counter Variables	61–3
Table 61-3	Counter, Timer, and Accumulator Structures	61–5
Table 61-4	Timer Structures	61–9
Table 61-5	Timer Variables	61–10
Table 61-6	Accumulator Structures	61–14
Table 62-1	OSI Structures	62–9
Table 62-2	Layer 1 OSI Variables	62–14
Table 62-3	Layer 2 OSI Variables	62–15
Table 62-4	Layer 3 OSI Variables	62–19
Table 62-5	Layer 4 OSI Variables	62–23
Table 62-6	Layer 5 OSI Variables	62–27
Table 62-7	Layer 6 OSI Variables	62–31

Table 62-8	Layer 7 OSI Variables	62–35
Table 63-1	Print Structures	63–2
Table 65-1	Status Structures	65–2
Table 66-1	Remote Port I/O Variables	66–3
Table 67-1	AUX Port I/O Variables	67–4
Table 68-1	Structure Fields—Other Library Tools	68–2
Table 68-2	Other Library Variables	68–4
Table 69-1	X.21 Structures	69–1
Table 69-2	X.21 Variables	69–3
Table 70-1	X.25 Layer 2 Structures	70–2
Table 70-2	X.25 Layer 2 Variables	70–3
Table 71-1	X.25 Layer 3 Structures	71–2
Table 71-2	X.25 Layer 3 Variables	71–3
Table 72-1	SDLC Structures	72–2
Table 72-2	SDLC Variables	72–4
Table 73-1	SNA Variables[	73–2
Table 74-1	DDCMP Variables	74–2
Table 75-1	Secondary Channel Structures	75–2
Table 75-2	Secondary Channel Variables[	75–3
Table 76-1	LAPD Structures	76–2
Table 76-2	LAPD Variables	76–3
Table 77-1	Q.931 Variables	77–2
Table 78-1	SS#7 Layer 2 Variables	78–2
Table 78-2	SS#7 Layer 1 Variables[	78–6
Table 79-1	SS#7 Layer 3 Variables	79–2
Table 80-1	High–Speed Frame Capture Variables	80–2
Table 81-1	ISDN S/T/U Structures	81–1
Table 81-2	U–Interface Message Type/Data Decodes for NT and LT State/Status Symbols	81–8
Table 81-3	S/T–Interface Message Type/Data Decodes for TE and NT State/Status Symbols	81–8
Table 82-1	SMDS SNI Structures	82–1
Table 82-2	SMDS SNI Variables	82–3
Table 83-1	ATM UNI Structures	83–1
Table 83-2	ATM UNI Variables	83–4
Table A4-1	Numbered Error Messages Returned for C Coding[	A4–1
Table C–2	Line Setup Clock Speeds	C–5
Table D1-1	Keyboard–to–EBCDIC	D1–2

Table D1-2	Keyboard–to–ASCII	D1–3
Table D1-3	Keyboard–to–EBCD	D1–4
Table D1-4	Keyboard–to–XS–3 (SYN=35; EOM=55)	D1–5
Table D1-5	Keyboard–to–IPARS	D1–6
Table D1-6	Keyboard–to–REVERSE EBCD	D1–7
Table D1-7	Keyboard–to–SELECTRIC	D1–8
Table D1-8	Keyboard–to–BAUDOT	D1–9
Table D1-9	Keyboard–to–JIS7	D1–10
Table D1-10	Keyboard–to–JIS8 (space parity)	D1–12
Table D1-11	Keyboard–to–JIS8 (mark parity)	D1–13
Table D2-1	Hex–to–Display Translation	D2–2
Table D3-1	Code–Set Characters	D3–4
Table D4-1	Keyboard–to–PC Conversion	D4–2
Table E–1	Remote Connector	E–2
Table E–2	Printer Connector	E–3
Table E–3	Auxiliary Connector	E–4
Table E–4	RGB Monitor	E–5
Table E–5	RS–232/V.24 Test Interface Module	E–7
Table E–6	V.35 Test Interface Module	E–9
Table E–7	X.21 Test Interface Module	E–11
Table E–8	RS–485 Test Interface Module	E–13
Table E–9	RS–449/V.36/V.37 Test Interface Module	E–15
Table E–10	ISDN Test Interface Specifications	E–17
Table E–11	ISDN S/T/U Test Interface Specifications	E–19
Table E–12	T1 Test Interface Specifications	E–21
Table E–13	Dual–Port T1 Test Interface Specifications	E–23
Table E–14	G.703 Test Interface Specifications	E–25
Table E–15	Dual–Port G.703 Test Interface Specifications	E–27
Table E–16	DS–0A 56K/64K Dual–Port Test Interface Specifications	E–29
Table E–17	G.703/64K Co–Directional Dual–Port Test Interface Specifications	E–31
Table E–18	TTL Test Interface Module	E–33
Table E–19	EIA–530 (DB–25) to EIA–449 (DC–37) Pin Comparison	E–35
Table E–20	V.35 Test Interface Connector Specifications (Dual–Port Mode)	E–36
Table E–21	V.35 Test Interface Connector Specifications (Single–Port Mode)	E–37
Table E–22	EIA–530 Test Interface Module (Dual–Port Mode)	E–38
Table E–23	EIA–530 Test Interface Module (Single–Port Mode)	E–39
Table E–24	EIA–232 Dual–Port Test Interface Connector Specifications (Dual–Port Mode)	E–40

Table E–25 EIA–232 Dual–Port Test Interface Connector Specifications (Single–Port Mode) E–41

Table H3-1 S1–Switch Settings for MPM Boards in the INTERVIEW 8000 Series H3–2

Part I

Basic Operation

1 Hardware

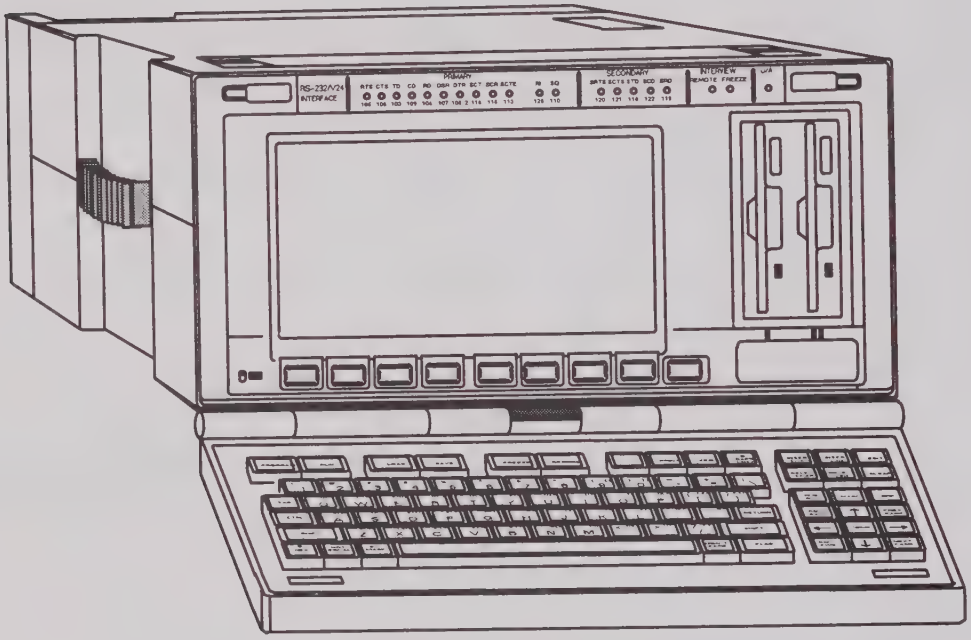


Figure 1-1 INTERVIEW 8800 *TURBO*.

1 Hardware

General Signal Networks offers a variety of INTERVIEW 8000 Series units, with varying capacity and multiplexer board/Test Interface Module configurations some of which may be optional for your unit. This manual addresses the hardware features common to all INTERVIEW 8000 Series units unless otherwise specified.

The INTERVIEW ATM units are described in the *INTERVIEW® 8800 PLUS ATM Technical Manual, Asynchronous Transfer Mode (Broadband).*

> **NOTE:** Genreal Signal Networks offers extended and expedited Maintenance Agreement plans for INTERVIEW 8000 Series hardware. Call Customer Service/Sales Administration for more information.

1.1 Physical Dimensions

The protocol analyzers in the INTERVIEW 8000 Series, represented by the INTERVIEW 8800 *TURBO* in Figure 1-1, measure 7 inches high by 14 inches wide by 18.5 inches deep (178 mm high by 356 mm wide by 470 mm deep). The unit weighs approximately 32 pounds (14.5 kilograms).

1.2 Keyboard

With the exception of R–Series *TURBO* units, the INTERVIEW has a 94–key keyboard containing ASCII keys and special keys separated into pads according to function. The keyboard is described in detail in Section 3.

The R–Series *TURBO* units do not come equipped with a keyboard since these remote units are controlled via a PC keyboard. The PC–to–INTERVIEW keyboard conversion appropriate for a particular remote application is contained in the documentation that accompanies the remote software.

1.3 Front Panel

(A) Remote INTERVIEWs

The front panels of the R–Series *TURBO*s have a HARD DISK indicator on the left that is red when there is activity on the hard disk.

There is also a single floppy–disk drive in the Drive 1 slot and indicator LED. See Section 1.4 for more information on floppy drives.

There is no display or keyboard on R–Series *TURBO* units. As the remote unit functions, the appropriate screens appear on the display of the controlling PC. An optional keyboard may be ordered (OPT–951–153–1) and a CGA monitor connected to use with the remote unit, if desired.

(B) Electroluminescent (EL) Display

The INTERVIEW's flat electroluminescent (EL) display screen measures 3.85 inches high by 7.69 inches wide (97.8 mm high by 193.3 mm wide). The high–resolution display (512 X 256 pixels) is black and yellow. No brightness adjustment is required for the EL display.

The screen has 21 display lines, 16 of which are devoted to data display. Data is displayed in lines 64 characters long, making the capacity of the screen 1344 characters (of which 1024 characters are devoted to line data). The top 2 lines of the data screen are devoted to status information; the bottom 3 lines to function key identification (see Figure 1-2).

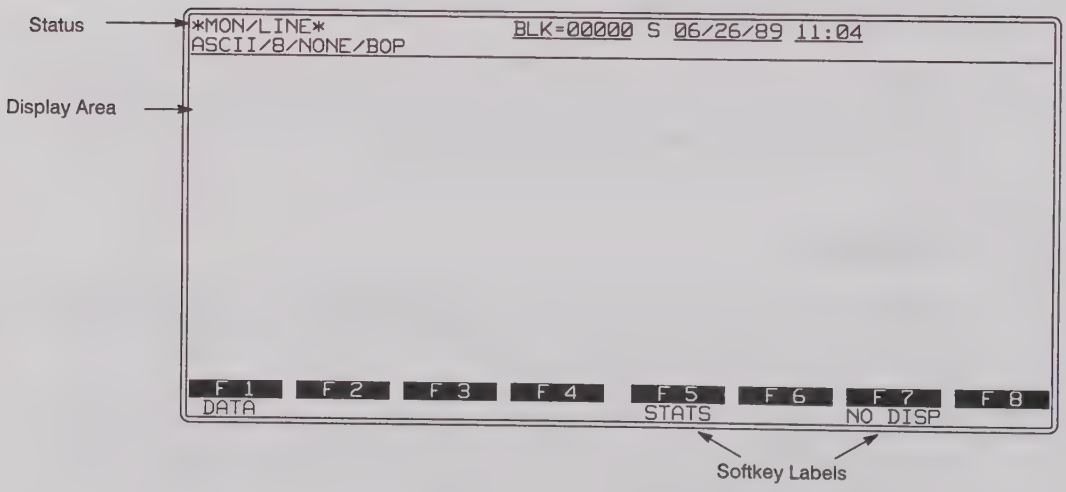


Figure 1-2 INTERVIEW display screen.

(C) Function Keys

Eight function keys and the DONE key are located directly below the display screen. The uses of the function keys vary from program menu to program menu; however, their function is always defined on the screen in rectangular windows located above the keys. (Refer to Section 3 for a discussion of their use.)

(D) LED's

Twenty LED's are placed above the display screen. These LED's are divided into three banks: interface status LED's, INTERVIEW status LED's, and the U/A LED. Figure 1-3 shows the assignment of LED's for an RS−232 interface.

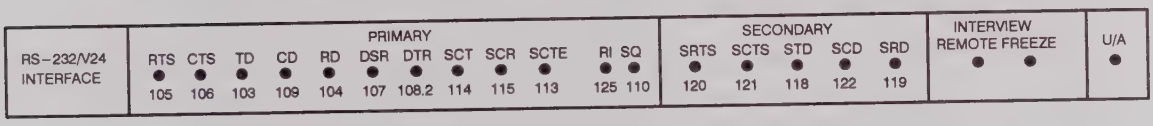


Figure 1-3 There are 20 LED's, divided by function, above the EL display.

1. *Interface status indicators.* Interface status indicators may be assigned to different signals, depending on the Test Interface Module which is installed in the rear panel. An overlay accompanies each module and should be placed over the front panel LED's whenever the module is installed. Assignment of each of the front panel indicators is illustrated in Appendix E, which also gives the interface specifications for each Test Interface Module.

   Primary and Secondary indicators on the front panel glow red to indicate that the lead is on (space voltage) and green to indicate that the lead is off (mark voltage). The eye recognizes orange when the leads are transitioning very rapidly. U/A lead indicators on Test Interface Modules operate differently. See Section 1.5(F).

2. *INTERVIEW status indicators.* There are three status indicators: REMOTE, FREEZE, and U/A. The REMOTE indicator is red when the INTERVIEW is under remote control. The FREEZE indicator is red when the display screen has been frozen (with the FREEZE key) while in Run mode. Both REMOTE and FREEZE LED's are dark when off.

3. *U/A LED.* The last indicator, U/A, is user−assigned and may be programmed to track any lead of the operator's choosing. See the RS−232 TIM documentation for the use of the UA−input jack in RS−232/V.24 testing.

(E) The Test Interface Overlay

Each Test Interface Module is accompanied by a front panel overlay. The overlay is placed over the interface status LED's and identifies the lead tracked by that LED when that particular Test Interface Module is installed. The overlay masks out any unused LED's. Replace the overlay each time that you change the Test Interface Module.

1.4 Disk Drives

(A) Floppy Disk Drives

The INTERVIEW uses 3.5 inch double−sided, high−density microfloppy diskettes (see Figure 1-4). Each formatted microfloppy has a storage capacity of 1.4 Mbytes.

Each disk has a write–protect window (see Figure 1-4). To write–protect a disk, slide the window open so that you can see through the disk.

If a high–density disk has been used on another piece of equipment which has a 1–Mbyte drive, it must be reformatted before it is used on the INTERVIEW. The data it contains cannot be read.

Disks from the INTERVIEW 5/10/15 PLUS Series can be read by units in the INTERVIEW 8000 Series when they have been properly formatted. Preparation of these disks and their use in the INTERVIEW is described in Section 14.

Like the remote 8000 Series *TURBO* units, the 8100 *TURBO* has one 3.5 inch microfloppy disk drive in the Drive 1 slot; the other 8000 *TURBO* Series units have two 3.5 inch microfloppy disk drives immediately to the right of the display screen. An LED just to the right of each drive is lit to indicate that the microfloppy in the drive is being accessed. Insert disks in the direction shown in Figure 1-5.

To remove a disk, press in the black bar next to the drive containing the disk.

> *CAUTION: Never remove a disk from its drive when the LED indicates that the disk is being accessed.*

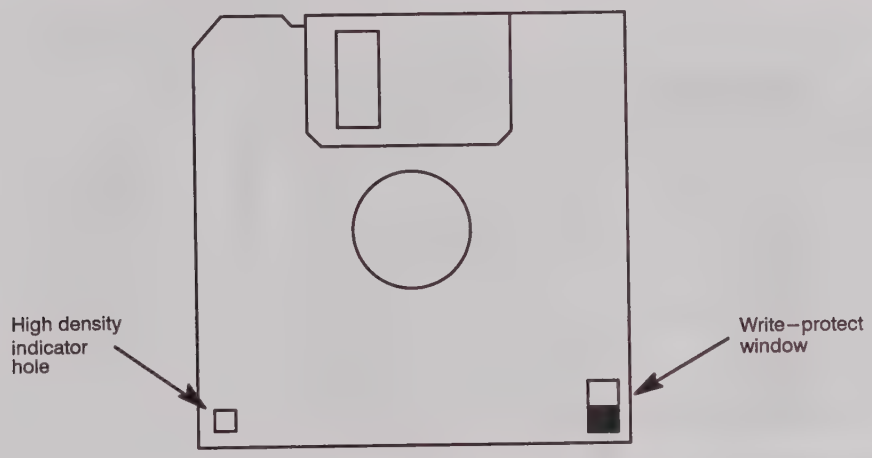


Figure 1-4 The INTERVIEW uses 3.5 inch quad– or high–density, double–sided microfloppy diskettes.

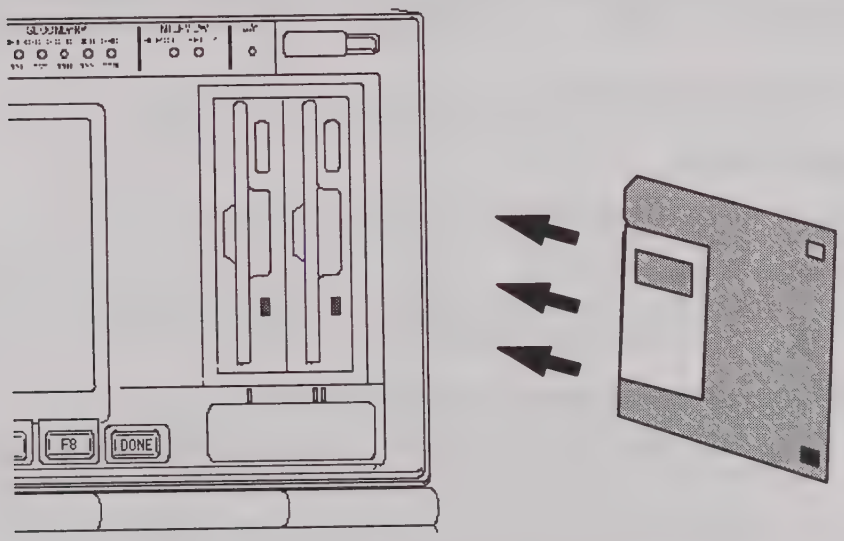


Figure 1-5 The INTERVIEW is equipped with two micro–floppy disk drives, except for the 8100 *TURBO* and R–Series *TURBO* units which have a single floppy drive; a SCSI hard disk is standard in all the 8000 *TURBO* Series units.

(B) Hard Disk Drive

All units in the INTERVIEW series are equipped with a SCSI hard disk drive. When the hard disk is being accessed, the indicator light on the left lower side of the front panel is lit.

Since the hard disk cannot be write–protected, you may wish to save its contents to microfloppy backup disks on a regular basis.

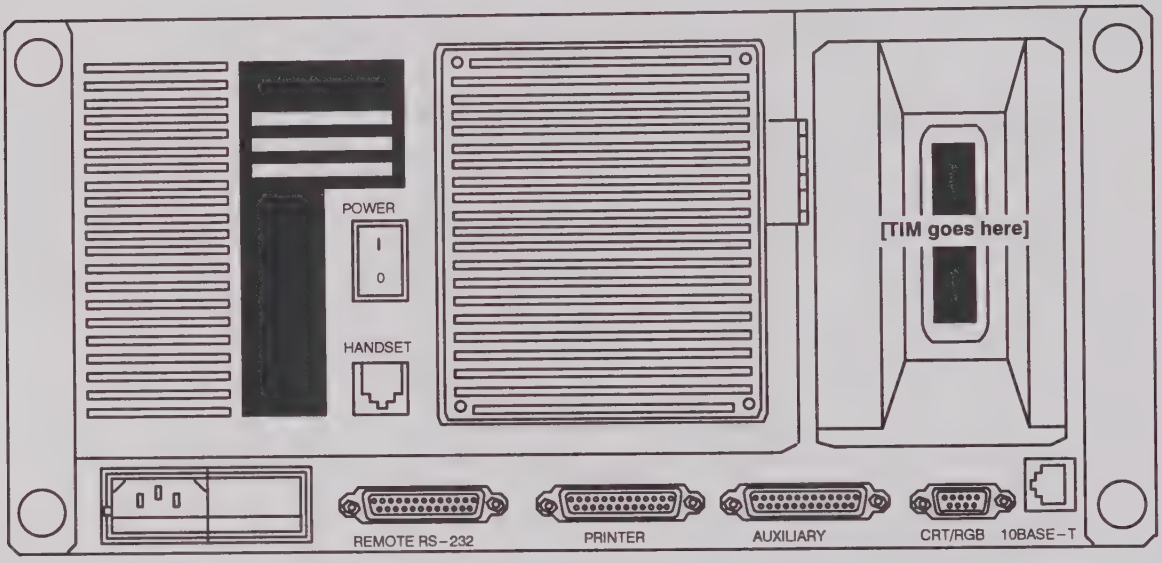


Figure 1-6 Back panel and connectors.

1.5 Back Panel

The back panel and its various connectors are shown in Figure 1-6.

(A) Power Module

The power connector is located at the bottom left of the rear panel. It is a standard three–wire grounded male connector, with selectable voltage.

(B) Voltage Selection and Fuse Replacement

Read the following information and instructions for the original INTERVIEW 8000 Series units or for the new ATM/ATM–ready INTERVIEWs.

1. *Original INTERVIEW 8000 Series units.* Original INTERVIEW 8000 Series units are designed to operate at 95 to 130 Volts ac when 115 V is selected or from 190 to 260 Volts ac when 230 V is selected. To determine the voltage currently selected in original INTERVIEW 8000 Series units, slide the transparent window of the power connector module to the left (see Figure 1-7). You will see the line voltage selector card at the bottom of the window. Current voltage selection is visible—and right side up.

   To change the line voltage selection, swing the fuse extractor handle (labeled FUSE PULL) out toward the left, and remove the fuse. The voltage selector card can then be removed and turned so the correct line voltage can be read right side up in the window. When the voltage selector card has been seated correctly, rotate the fuse extractor handle to the right and in, and replace the 4 amp fuse.

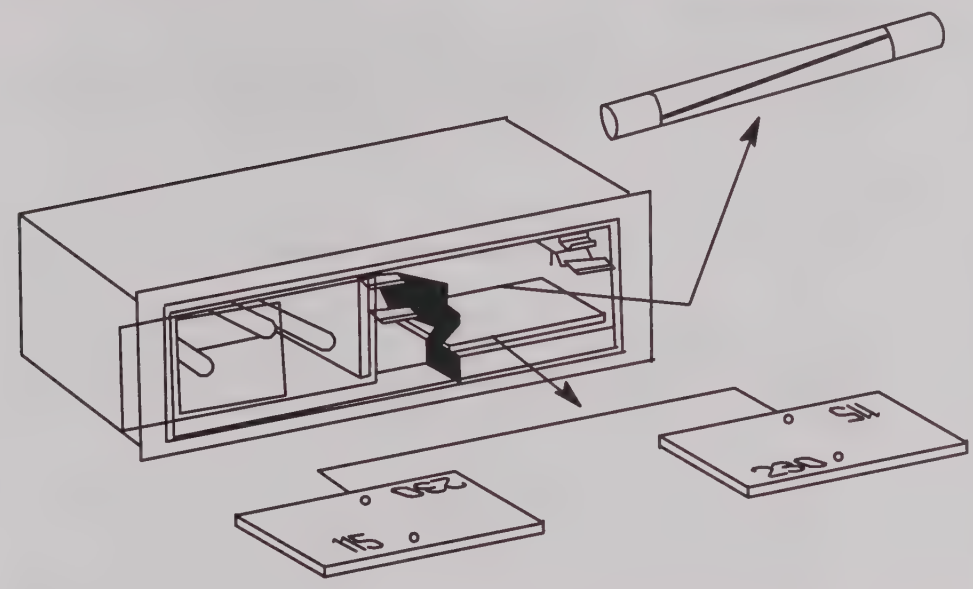


Figure 1-7 For original INTERVIEWs, voltage is selectable in the power connector module with a voltage card; in newer, ATM–ready INTERVIEWs, the required voltage is sensed and selected automatically, with no voltage card required.

Included in your shipment is a detachable power–supply cord with a NEMA 5–15 attachment plug rated 15 A, 125 V. If you configure the unit for 220–240 V operation, you should employ a UL–listed (or appropriate approval for your locale) power–supply cord set furnished with a grounding plug suitable for connection to the 220–240 V source of supply.

The unit will operate the display at either 50 or 60 Hz refresh rate. It defaults to 60 Hz *unless* a file named */sys/fifty_hertz* is listed on the boot–drive disk. If a file with this name is created, the unit will operate at 50 Hz. (The content of the file is irrelevant and will be ignored by the boot–up software.)

2. *ATM–ready INTERVIEWs.* ATM and ATM–ready INTERVIEWs are designed to operate anywhere in the range from 95 to 260 Volts ac. These units do *not* require a voltage selection card; the power supply in these units senses and selects the correct voltage.

   To change the 5 amp fuse, refer to Figure 1-7, but note there is no voltage card present in the new models. Swing the fuse extractor handle (labeled FUSE PULL) out toward the left, and remove the old fuse. Rotate the fuse extractor handle to the right and in, and replace the old fuse with a new 5 amp fuse.

   As before, a file named */sys/fifty_hertz* should be listed on the boot–drive disk if the unit is to be operated at 50 Hz, as described in subsection 1., above.

(C) On/Off Switch

The power switch is located above the power connector and to the right. Press the side of the switch marked "1" to turn power on. Press the side marked "0" to turn power off.

(D) The Fan

The INTERVIEW is cooled by a fan which may be accessed through the rear panel.

A fan filter prevents dust and dirt from getting into the INTERVIEW. As the filter gets dirty, less cooling air gets to the unit. To prevent your INTERVIEW from overheating, we recommend that you periodically clean the filter. A plastic grill on the rear of the unit covers the fan filter.

> *CAUTION: Do not insert objects through the grill covering the fan. Do not remove the grill without turning off the unit and disconnecting power.*

Turn the INTERVIEW off, disconnect the power, and remove the screws holding the grill in position. Remove the filter, rinse it in clean water, dry it thoroughly, and replace it. Screw the grill back on the the unit.

If your INTERVIEW is overheating and cleaning the filter does not alleviate the problem, contact Customer Service.

(E) Connectors

The following is a brief description of all I/O connectors on the rear panel of the INTERVIEW. Interface specifications for each of the connectors are given in Appendix E.

1. *Handset connector.* The ISDN Handset Connector is just below the power switch. It is a standard RJ–11C connector. This interface is intended for an optional ISDN handset.

2. *Remote RS–232 connector.* This is an RS–232/V.24 25–pin connector located just to the right of the power connector. It provides access to an external modem (or directly to another INTERVIEW unit).

3. *Printer connector.* The RS–232/V.24 25–pin printer connector is located directly to the right of the Remote connector. It allows access to most serial printers. The connector acts as DCE and transmits on RD. Printer operations are described in Section 16.

4. *Auxiliary connector.* The Auxiliary I/O connector is a 16−bit bidirectional TTL connector which allows access to external peripheral devices. Use the C routines discussed in Section 67 to control and monitor this interface. Other references to AUX leads in this manual pertain to the four AUX pins on the RS−232 TIM (see Figure E−1).

   CAUTION: Never plug an RS−232/V.24 cable into the Auxiliary connector, as the signal voltage is likely to damage the interface.

5. *CRT/RGB connector.* This is the color video connector. Signals from the INTERVIEW display can be passed through this connector to a color monitor to produce color graphics and other displays in color. Color, vertical sync, horizontal sync, and intensity signals can be controlled from the external monitor. Use of color in displays is described in Section 18.

6. *10BASE−T connector.* Each Ethernet−capable INTERVIEW unit is equipped an RJ−45 connector for making 10BASE−T ethernet connections.

   NOTE: Ethernet functionality requires both operating system software version 14.00 or greater ***AND*** firmware revision 10.00 or greater. (OS version 14.00 will not boot on firmware with revision levels *below* 10.00.)

   NOTE: INTERVIEW units produced prior to December 1996 which have not been upgraded for Ethernet capability come equipped with a *composite video connector* instead of the RJ−45 connecor in the lower right−hand side. This connector provides RS−170 video output to an external monitor or camera.

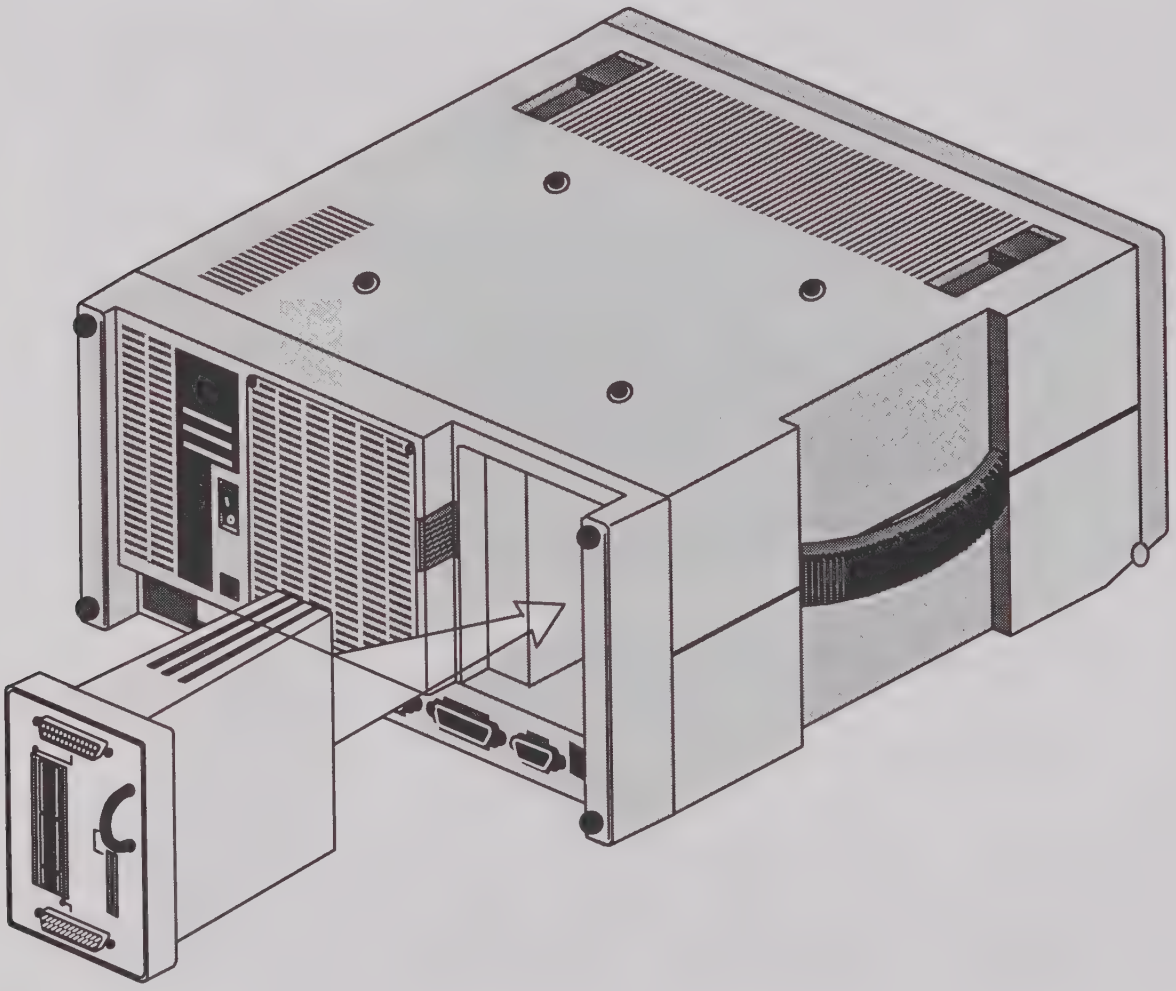


Figure 1-8 The right rear panel accommodates interchangeable Test Interface Modules.

(F) The Test Interface Module

The empty receptacle to the right of the rear panel (see Figure 1-6) accommodates interchangeable Test Interface Modules (TIM's). Whenever the INTERVIEW monitors a data line or emulates a DTE or DCE, the correct TIM must be installed. At least one customer–selectable interface is standard with any unit in the INTERVIEW series (except for the INTERVIEW 8750 ATM *EXPRESS*, which does not have WAN capability). Other test interfaces are available as options; see Section 13. Most Test Interface Modules are equipped with two connectors, a TO DTE and a TO DCE connector, with the dual–port series available with connectors for each port. Figure 1-8 shows a single–port RS–232 module being inserted into the unit.

Connect to the data line as described in Section 1.10. The interface softkey (F2 on the Setup Menu) will reflect the type of TIM installed, such as RS232, V.35, RS449, RC8245, TTL, etc.

CAUTION: To connect the data line, you must interrupt the flow of data on the line. Be sure you have permission to break the line before doing so. General Signal Networks' DATA–PATCH® systems and AUTONEX matrix switch systems often provide non–intrusive access methods for monitoring the line.

NOTE: It is possible to monitor data previously recorded on disk, whether or not any TIM or the correct TIM is installed.

1. *Software control of TIM connectors.* When **Mode:** MONITOR or AUTOMON is the program selection, the INTERVIEW passively monitors data through either (or both) TO connectors on the Test Interface Module.

   When the INTERVIEW is operating in EMULATE DCE or BERT DCE modes, the TO DTE connector is active. The INTERVIEW is transmitting and receiving data through the TO DTE connector. When **Mode:** EMULATE DTE (or BERT DTE) is the program selection, the INTERVIEW transmits and receives data through the TO DCE connector. The interface specifications for each Test Interface Module are given in Appendix E.

   Break–out switches on each Test Interface allow any pin to be patched. See the RS–232 TIM documentation for an explanation of the RS–232 breakout switches.

2. *Test interface LED's.* There are four LED's on Test Interface Modules which do not require a MUX board. Two, labeled EMULATE DTE and EMULATE DCE, indicate the operating mode of the unit. When EMULATE DCE is red, the TO DTE connector is active; when EMULATE DTE is red, the TO DCE connector is active. When the INTERVIEW is monitoring or auto–monitoring, both EMULATE LED's are black.

   The two LED's above the U/A input on the patch panel track the voltage level on the lead patched to U/A. The red LED above the U/A panel is lit to indicate space voltage (positive voltage above a minimum threshold). When the green LED above this panel is lit, it indicates a mark voltage (negative voltage within a specified acceptable range). For intermediate voltages, the U/A LED's are off (see Figure 1-9).

   CAUTION: Power off the INTERVIEW before installing or removing a TIM.

   Remember to change the front panel overlay each time that you change the TIM.

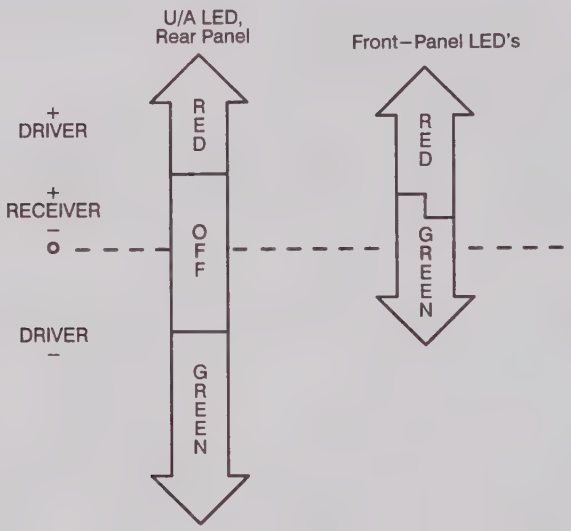


Figure 1-9 Color phases in green–red LEDs.

1.6 Storage Capacity

(A) RAM

RAM capacity for line data in the screen buffer is 64 Kbytes—a maximum of 32 Kbytes of characters plus 32 Kbytes of attributes for character data. EIA leads and time ticks are stored in RAM with the characters, if these options were selected on the Front–End Buffer menu (see Section 9.1). Each byte devoted to EIA leads and time ticks reduces the number available for characters. Bit–image RAM is 256 Kbytes in the 8100 *TURBO and* 8200 *TURBO* Series units and 1 Mbyte in the 8600 *TURBO* Series, 8700R *TURBO* Series, and 8800 *TURBO* Series units. (Bit–image RAM may be increased. See Section 7.5.)

(B) Microfloppy Diskettes

Each diskette has a 1.4 Mbyte storage capacity (formatted); thus, total diskette capacity is 2.8 Mbytes for the INTERVIEW 8200 *TURBO*, 8600 *TURBO*, 8700 *TURBO*, and 8800 *TURBO* units, and 1.4 Mbytes for the INTERVIEW 8100 *TURBO* unit with its single floppy disk drive. (An optional floppy disk drive will increase storage capacity (formatted) to 2.8 Mbytes for the INTERVIEW 8100 *TURBO* unit.)

(C) Hard Disk Drive

The hard disk drive has a storage capacity equivalent to the size of the SCSI disk drive installed in the unit (e.g., 240 Mbytes with 240–Mbyte SCSI drives, 2.1 Gbytes with 2.1–Gbyte SCSI drives).

(D) Maximum Data Rates

Maximum rates for data recording are as follows:

- Recording to bit–image RAM
  (full–duplex, 100% line use): 2.048 Mbits per second

- Bit–image recording to 80–Mbyte (Winchester) hard disk
  (full–duplex, 100% line use): 768 Kbits per second

- Bit–image recording to SCSI hard disk
  (full–duplex, 100% line use): 2.048 Mbits per second

- Recording to microfloppy
  (full–duplex, 100% line use): 64 Kbits per second

The speed your INTERVIEW actually achieves may vary. Factors which will influence data–analysis rates are line speed, percentage of line utilization, average frame length, the layer packages and user program loaded, suppression of idle, and the time–stamp resolution. Refer to Sections 2.10 and 2.11 on how to optimize the INTERVIEW's speed of operation.

1.7 Clock

Data clocking is provided by a set of high–frequency crystals. The various data speeds that can be selected are listed in Appendix C.

The INTERVIEW is also equipped with a time–of–day clock which provides readings of time (hours, minutes, seconds), day, month, and year. Time may be used as an INTERVIEW program condition. Refer to Section 17 for instructions on setting the time–of–day clock. Refer to Sections 25 and 31 for a description of Time as a program condition.

1.8 Operating Environment

The INTERVIEW is designed to operate in an atmospheric temperature ranging from 41 to 122 degrees Fahrenheit (5 to 50 degrees Celsius). At these temperatures, the unit can operate in (uncondensed) humidity ranging from 30 to 90 percent.

NOTE: Operation at extreme temperatures require the fan filter be clean. See Section 1.5(D).

CAUTION: Avoid dropping the unit.

Avoid getting the unit wet.

Do not operate the unit with the fan covered.

Avoid operating the unit immediately after exposure to drastic changes in temperature and humidity.

Avoid placing the INTERVIEW on a radiator or near a source of heat.

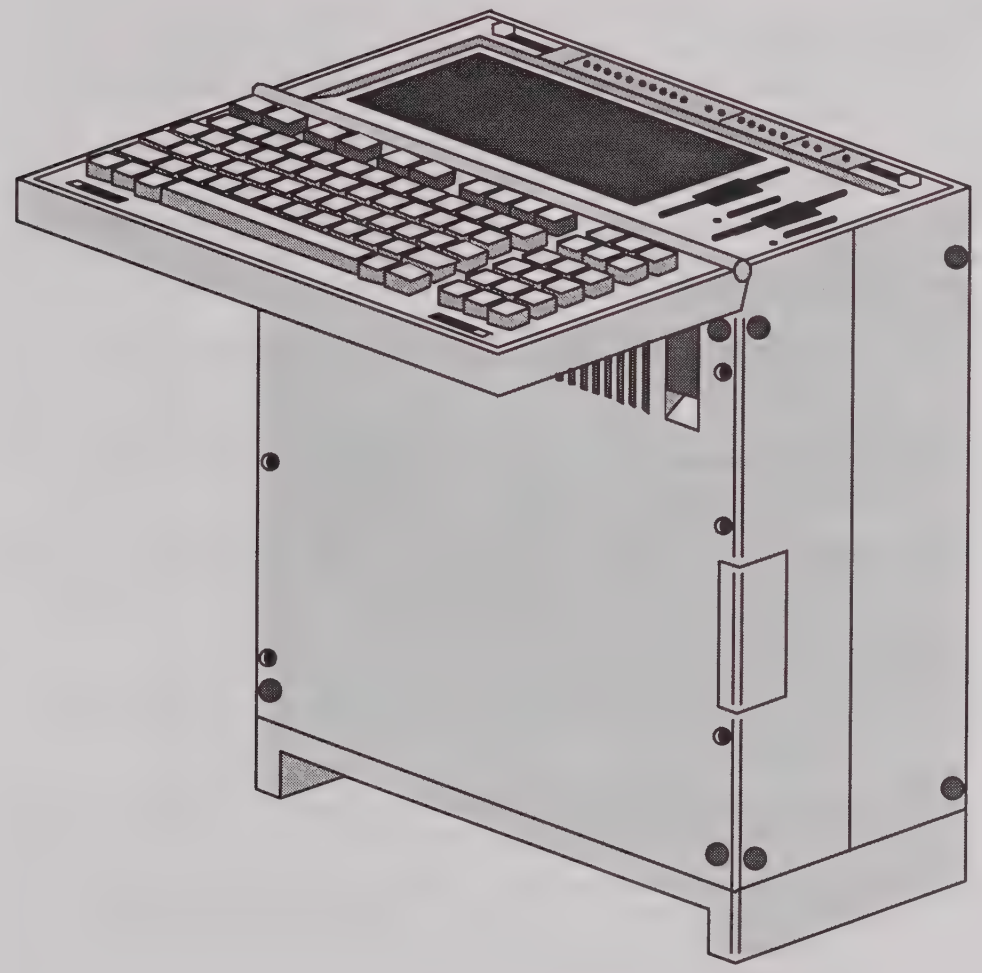


Figure 1-10 The INTERVIEW may be operated on a desk top or in a standing position.

1.9 Operating Positions

The INTERVIEW is designed to operate on a desktop (Figure 1-1) or in a standing position, with the display facing upwards (Figure 1-10). The unit may also be shelf–mounted in an equipment rack or cabinet. Allow for adequate air flow when mounting the INTERVIEW in a rack.

CAUTION: To protect the hard disk, do not move the unit with the power on. Turn the power off first to position the hard disk in a protected state. However, do ***not*** *turn the power off when the disk–access LEDs are* ***on****.*

1.10 Power Up

Before you power up the INTERVIEW, make sure you have performed the preliminary steps listed below.

(A) Install the Test Interface Module

Check the rear panel to be certain that you are using the correct Test Interface Module before you test any data. A Test Interface Module is shown in Figure 1-8 and described in Section 1.5(F).

> *CAUTION: Never install or remove a TIM unless the INTERVIEW is powered off.*

Remove the Test Interface Module by pressing the button to the left of the module with the thumb of your left hand. Hold the button in as you pull firmly on the TIM handle with the thumb and forefinger of your right hand.

Install the Test Interface Module right–side–up in the receptacle provided at the right of the rear panel. Press firmly on the top and bottom of the module until it is secured and the button to the left clicks into place.

(B) Insert the Correct TIM Overlay

Be certain that the overlay over the front–panel LED's matches the Test Interface Module installed.

To remove the overlay, grasp the top and bottom edges with the thumb and forefinger of both hands and bow the strip slightly in the center. The overlay should slip out.

To insert an overlay, bow it slightly. Insert the tabs on either side of the overlay into the notches on each side of the LED panel. Make sure that the holes on the underside of the overlay are placed over the small posts at the top of the empty LED panel, and press the overlay into place.

(C) Connect to a Data Source

It is not necessary to install a Test Interface Module or connect to a data line if you are reviewing data stored on disk.

If you plan to test a data line, connect to the line as described below.

> *CAUTION: You must interrupt the flow of data when you connect to a data line. Be sure you have permission to break the line before doing so.*

Break the data line for testing. For single–port testing, connect one end of the line to the TO DTE connector on the Test Interface installed in the rear panel; connect the other end of the line to the TO DCE connector on the TIM. For dual–port

testing, DTE and DCE are on the same connector for each port; connect each port on the TIM to the respective line to be tested. Even when the INTERVIEW is powered off, this provides a through connection for the data line.

An LED next to the connector will be lit if the INTERVIEW is actively testing—or is programmed to test—on that connector.

Refer to Section 1.5(F) for a description of TIM connector functions.

Check the voltage selection on the card in the small window on the rear panel.

If the voltage is incorrect, refer to Section 1.5(B) for instructions.

(D) Open the Keyboard

Place the INTERVIEW on a stable surface. Support the back of the keyboard with one hand. Unlatch the keyboard by pushing the blue latches on the top of the unit all the way back. Then lower the keyboard to operating position.

(E) Power On

Connect the female end of the power cord provided to the back of the unit; connect the other end to a standard power outlet. Depress the side of the power switch marked "1." A Start Up screen similar to that in Figure 2-1 should appear. Refer to Section 2 for a discussion of the Start–Up screen, system initialization, and general operations.

2 General Operation

This section discusses the general operation of the INTERVIEW and covers such topics as updating system software, configuring the menus, starting and stopping a test program, locating errors detected when the test is compiled, using both the preexistent trigger setup menus and the free−form spreadsheet to create test programs, and using the INTERVIEW's analysis features.

For a hands−on introduction to the INTERVIEW, see *30 Minutes to Programming the INTERVIEW 8000 Series*, 951−B0431−01.

2.1 Power Up

As you power up the INTERVIEW, perform the preliminary steps listed below. The procedures for each step are described in Section 1, Hardware.

- Install the Test Interface Module.
- Select the correct voltage, if required.
- Open the keyboard.
- Insert the proper TIM overlay.
- Install the initialization disk if you are not booting from hard disk.
- Connect the power cord and turn power on.
- Connect to a data source.

(A) Self Tests

When you turn on the unit, you initiate a series of self tests: first, the CPM Module DRAM, then the MPM Module DRAM, and finally, the MPM to CPM connections. Any self−test errors will be reported on this screen. Refer to the Appendices for an explanation of error messages. You may abort the self tests by pressing PROGRAM. (However, we recommend that you allow the tests to run their course.)

Once the self test cycle is complete or once you have aborted the tests, the INTERVIEW begins to initialize its software. The message BOOTING appears on the screen. The default drive at power−up is the disk from which the system loads initialization software. The INTERVIEW first checks FD1, then FD2, and finally, HRD. As soon as it detects system software, it stops the search and boots up.

NOTE: Operating system software version 14.00 or greater requires firmware revision 10.00 or greater to run and vice versa. If you attempt to operate OS version 14.00 on a unit with firmware at a revision *below* 10.00, the unit will not boot up.

NOTE: You can reset the INTERVIEW directly from the keyboard. Press the CTRL–SHIFT–DELETE CHAR or CTRL–SHIFT–DELETE LINE key combination to force the INTERVIEW to perform the self–test and booting sequences without turning the unit off. This reset is similar to that of a PC in that it is not a hardware reset. For hardware resets, you must power the unit off and then back on.

When self tests are complete, a Start Up screen similar to that in Figure 2-1 will appear. Notice that when you powered on, all the following information appears on this screen: the types of disk drives installed, the number and types of processors in your unit, the software and firmware versions, the options installed, the Test Interface Module in place, and the optional TIM crystal installed (if any).

```
          ** INTERVIEW 8700 TURBO **

      DISKS: FLOPPY 1  FLOPPY 2  HARD DISK (240M)

      PROCESSORS: 4 (1:68010-8;3:80286-12.5)

      SELF-TEST ERRORS: NONE

Press:
[PROGRAM] to enter the menu page
[RUN]     to run the default program

Software Version: 14.00
Firmware Version: 10.00

OPTIONS: Mux

TIM: RS-232/V.24
TIM CRYSTAL: 6144000
Copyright (c) 1987, 1996
GENERAL SIGNAL Networks
```

Figure 2-1 Power–up screen, INTERVIEW 8700 *TURBO*.

(B) File Searches

During power–up, the INTERVIEW searches specific directories for the files described below.

1. *print_setup.* During boot–up, the */sys* directory on the boot drive is searched for the file *print_setup*. If it exists, the configured Printer Setup in this file is automatically loaded. Otherwise, the INTERVIEW loads the default Printer Setup. See Section 16.3.

2. *user_intrf.* If you have a file named *user_intrf* in the */usr* directory on the hard disk, the program in *user_intrf* will be compiled and run as soon as boot–up is complete, bypassing the Start Up screen as shown above. See Section 2.2(A).

3. *default.* The Start Up screen will appear only briefly if you have a file named *default* in the */usr* directory of your boot–up disk. If you do not press PROGRAM within five seconds of power–up, the program in *default* will be compiled and run. See Section 2.2(B).

4. *ezview_setup.* During boot–up, the INTERVIEW tries to locate the file *ezview_setup* in the */sys* directory on the boot–up disk. The INTERVIEW's Easy View operation is configured according to the parameter settings in this file. See Section 19. If the file is not located, the INTERVIEW is configured with all default YES selections on the Easy View Setup menu. If this menu shows the selection **Enter Easy View After Power–Up:** YES, and neither *user_intrf* nor *default* is located, the Start Up screen appears for five seconds. Then, the first menu in the Easy View system is displayed. See Section 4. (If *user_intrf* or *default* is located, the unit does not enter Easy View following boot–up regardless of the setting of this parameter.)

2.2 Initializing System Software (Booting Up)

The INTERVIEW 8000 *TURBO* Series units normally do not require a system disk, since all initialization software is installed on the hard disk. If you need to install new system software from a floppy disk, refer to Section 2.3.

> **NOTE:** We recommend that you make a copy of system software to use as your boot disk. Use the Duplicate Disk command on the Disk Maintenance screen as explained in Section 14.4(D). Then store your original floppy in a safe place.

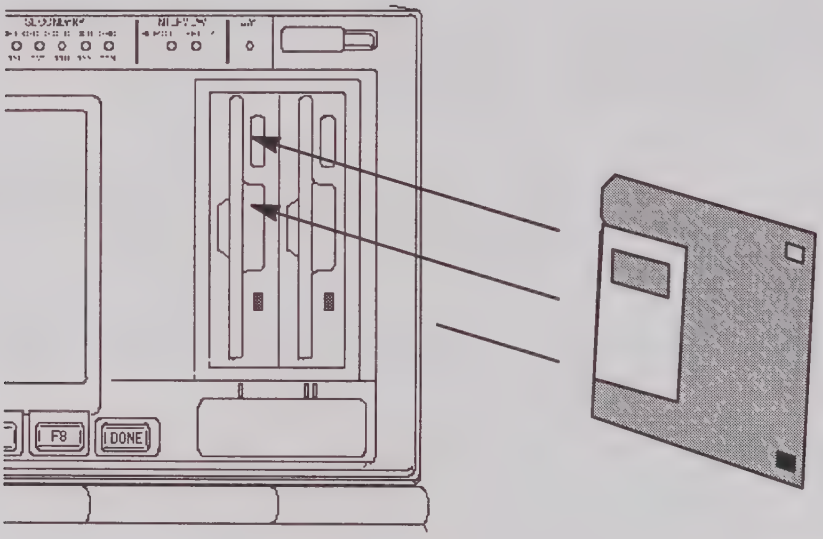


Figure 2-2 Insert microfloppy disks as shown.

(A) Creating a User Interface

Regardless of which disk drive you use for boot–up, FD1, FD2, or HRD, the *HRD/usr* directory—and only that directory—is searched during power–up for a file named *user_intrf.* If the file is located, the unit will automatically load, compile, and run the program as soon as boot–up is complete. Each time the operator presses PROGRAM, the program in *user_intrf* will be loaded, compiled, and run again. Use this feature when you want to bypass the INTERVIEW's menus and create your own user interface for specific applications.

> *CAUTION: Avoid saving emulation programs in* user_intrf. *Booting up and automatically running an emulation program may result in an inadvertent break of the line.*

To enter Program mode, press SHIFT – PROGRAM or CTRL – PROGRAM. Perform any Program–mode operation you wish: make selections on menus, execute File or Disk Maintenance commands, or create a Protocol Spreadsheet program. To enter Run mode again, press RUN. Each time you execute the PROGRAM key during Run mode, *user_intrf* will be loaded, compiled, and run.

To prevent the program in *user_intrf* from automatically running upon power–up, change the name of the file. Simply capitalizing the first letter in the file name *(User_intrf)* is sufficient. If *user_intrf* is not found during power–up, the PROGRAM key cannot be used to enter Run mode, even if *user_intrf* is saved to the *HRD/usr* directory and manually loaded, compiled, and run. Turn off the INTERVIEW and power up again to activate the user–interface feature.

NOTE: References to the [PROGRAM] key throughout this manual assume that the *user_intrf* file was not located during power–up. In other words, the [PROGRAM] key is used to enter Program mode only.

To create a *user_intrf* program:

1. Configure the menus to the selections you want.
2. Use Protocol Spreadsheet softkey entries or C regions on the spreadsheet to develop your user–interface program. All of the C structures, variables, and routines available to the INTERVIEW user are explained in Part II of this manual, *Advanced Programming.*
3. Press [SAVE] to call up the File Maintenance screen.
4. Check the hard disk for any existing user–interface program. Press [F3] for **Command:** CHANGE DIR. Select **Drive:** HRD. After **Name:** type in */usr* and then press [XEQ]. Check for a file named *user_intrf*.

   Unless the old file is write–protected, any program already stored under this filename will be overwritten when you save your new file. To keep the old file for later reference, save it to a new name (its contents will have to be loaded manually). For a detailed discussion of file–maintenance commands, see Section 15.
5. Select **Command:** SAVE. Select **Type:** OBJECT or PROGRAM. Then select the hard disk.

   NOTE: The more complex a program is, the longer it takes to compile. To eliminate compilation each time you use the [PROGRAM] key, therefore, we recommend that you save *user_intrf* as an object file.

6. In the **Name:** field, type in the filename */usr/user_intrf*. Only this name can be used. (Program or object files saved to any other name or directory must be loaded manually.)
7. Execute the SAVE command by pressing [XEQ].
8. *user_intrf* will appear in the Directory Listings when *HRD/usr* is the current directory. (The name of the current disk appears as a prefix to the absolute pathname of the current directory. The name of the current directory appears on the fourth line of the File Maintenance screen.)
9. You may alter the *user_intrf* program again at any time by saving a new program to the same filename.

(B) Running the Default Program

When the INTERVIEW boots up, the */usr* directory on the boot–up disk is searched for a file named *default.* Once the start–up screen (Figure 2-1) appears, the program in *default* (if it exists) will be compiled and run automatically after five seconds, or immediately if you press [RUN]. You may prevent the default program from running by pressing [PROGRAM] before the five–second timeout expires.

Develop a default program to suit your particular needs. One application of the default program might be defining a new set of default menu selections.

> *CAUTION: Avoid saving emulation programs in* default. *Booting up with a default emulation program may result in an inadvertent break of the line.*

Upon boot–up, all menu selections in the INTERVIEW are set to certain values. You may change these default selections if you wish by utilizing the default program. Follow these steps:

1. Configure the menus using the default selections you want.
2. Press [SAVE] to call up the File Maintenance screen.
3. Check your disk for any existing default program. Press [F3] for **Command:** CHANGE DIR. Select the boot disk you want to use. After **Name:** type in */usr* and then press [XEQ]. Check for a file named *default*.

   Unless the old file is write–protected, any set of defaults already stored under this filename on the disk you have selected will be overwritten when you save your new default file. To keep the old file for later reference, save it to a new name (its contents will have to be loaded manually), or use a different disk for your new defaults. For more assistance, refer to Section 15.
4. Select **Command:** SAVE. Select **Type:** PROGRAM, SETUP, or OBJECT. Then select the disk from which the system will be initialized.
5. In the **Name:** field, type in the filename */usr/default*. Only this name can be used as the new set of defaults. (Program files saved to any other name or directory must be loaded manually.)
6. Execute the SAVE command by pressing [XEQ].
7. The name of the file you have saved will appear in the Directory Listings when */usr* is the current directory and the selected disk is the current disk. (The name of the current disk appears as a prefix to the absolute pathname of the current directory. The name of the current directory appears on the fourth line of the File Maintenance screen.)

8. You may alter these defaults again at any time by saving a new default program to the same filename.

9. Be sure that you initialize the INTERVIEW from the disk which contains *default*. The unit will load, compile, and run the default program automatically (unless you press [PROGRAM]). The Run–mode screen displayed will be the one selected in the Display Setup menu in *default*. Likewise, all other menus will reflect your customized defaults.

   Refer to Section 15 for more information on the filing system or file maintenance commands.

2.3 Installing New System Software on Hard Disk

From time to time, you may need to install new system software on the hard disk of an INTERVIEW 8000 *TURBO* Series unit. Use the Duplicate Disk command from the Disk Maintenance utility. (See Section 14.4(D) for more information on this command.) The steps are as follows:

1. Write–protect the master copies of the new system, user, and ACE Package disks. Slide the plastic tab so that you can see through the rectangular write–protect hole.

   NOTE: There should also be a second rectangular hole which does not have a sliding tab. If you have any 3.5 inch disks which do not have this second hole, they are not compatible with the INTERVIEW.

2. Insert the new system floppy disk (DSK–951–001–1.X) into Floppy Drive 1 (FD1), the left–hand floppy drive.

3. From the Main Program menu, press the UTIL softkey to bring up the Utility menu, then D/MAINT for Disk Maintenance. Select the softkey labeled DUPDISK to access the Duplicate Disk command screen.

   NOTE: If you are in Easy View (see Section 4), first press [EZ VU], then [PROGRAM] to access the Main Program menu.

4. Select From Disk Number: FD1 and To Disk Number: HRD on the command screen. Then press [XEQ]. The system will prompt you to insert a disk. Since the system disk is already in FD1, press the [F1] softkey (GOAHEAD). When the duplication is completed, the system will prompt you again to insert the next disk. Remove the system disk from FD1 and insert the user floppy disk (DSK–951–001–2.X) for duplication. Press [F1]. Then repeat the process with the ACE Package disk (DSK–951–001–7.X).

   The only files overwritten on the hard drive will be the system software files in the */sys* directory and files from the */usr/layer_pkgs* directory. These files comprise the new system software.

5. Once copying is complete, take the master copy of the user disk out of Drive 1, and store it and the master copy of the system disk in a safe place.

> **NOTE:** We recommend that you make a working copy of the new software on floppy disks which can be kept with the INTERVIEW.

6. Turn off the power switch for the unit, and wait ten seconds. Then turn the power on again to reboot the INTERVIEW. Following the self–test, the unit should boot without error. The new software version should appear on the screen. If there are errors or the unit will not reboot, repeat Steps 2 through 6. If problems persist, contact Customer Service/Technical Support.

2.4 Backing Up the Hard Disk

Periodic back–up of the hard disk is strongly recommended.

1. Install a formatted diskette in Drive 1. This disk should not contain operating system software. For formatting instructions, see Section 14.4(A).

2. Go to the File Maintenance screen. (From the Main Program menu, press F6.) Use the File Maintenance Copy command to copy any files you wish from the hard disk to floppy.

> **NOTE:** Do not copy files of type SYS or any files from the */sys* directory or the */usr/layer_pkgs* directory to the backup disk. These files reside on the master copies of your system and user software disks.

3. Once copying is complete, take the backup disk out of Drive 1 and write–protect it. Slide the plastic tab so that you can see through the rectangular write–protect hole. Store the disk in a safe place.

Select from the following methods if you need to recopy files from the backup disk to the hard disk. *Keep in mind that files on the hard disk with the same name as those on the floppy will be overwritten.*

- Copy files or directories one at a time using the File Maintenance Copy command.
- Copy the root directory from the floppy to the root directory of the hard disk. The name to enter for the root directory—once you have selected the correct origin or source drive in the rotating field—is simply the slash character, /.
- Use the Dupdisk command on the Disk Maintenance screen to duplicate the contents of the backup disk on the hard disk.

NOTE: Do not recopy files from the */sys* directory, files whose type is SYS, or files from the directory */usr/layer_pkgs*. If you need to reinstall these files, use the master copies of system and user software disks to avoid inadvertently overwriting more recent software version files with older ones.

2.5 The Menus

The INTERVIEW is used to monitor data as it is received through a data line or to playback and monitor data as it was recorded from the line. The INTERVIEW may also be set up to emulate one side of a communication, sending data and responding to the data it receives. A series of menus are used to set the unit for the data you expect to receive or send and the type of analysis you wish to perform. These menus are categorized on the main Program menu. Select one of these menus by using the [ROLL FWD] or [ROLL BACK] key, or by pressing the labeled softkey. Enter the menu by pressing the highlighted function key or [RETURN].

(A) The Program Menu

Press [PROGRAM] to see the Program menu. Notice that the Software and Firmware Versions are posted at the top of the screen. They are available to you whenever you return to this menu.

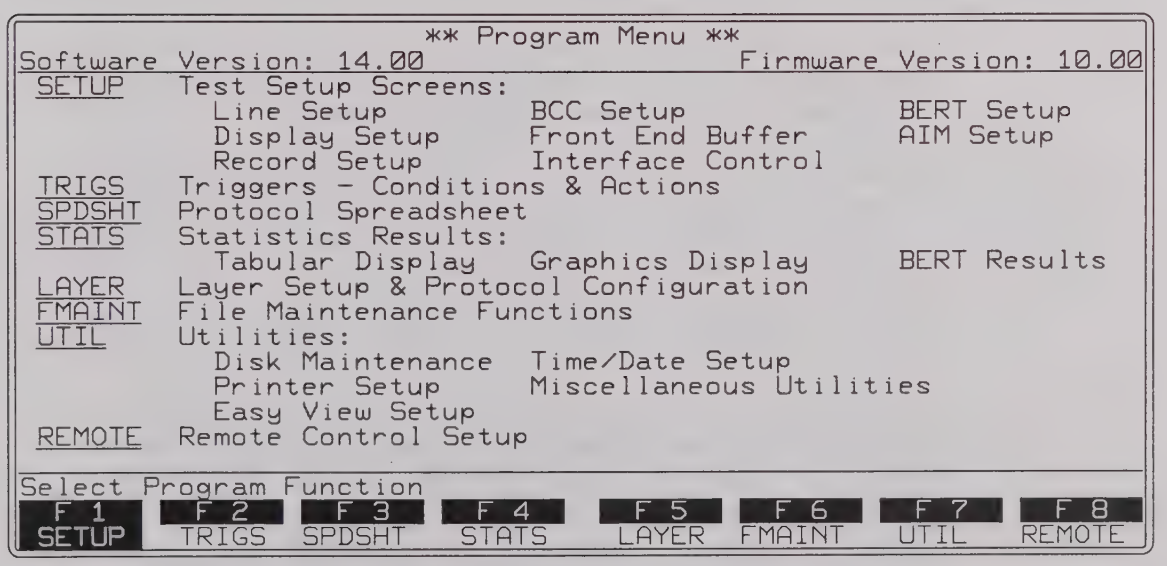


Figure 2-3 Main Program menu.

Think of the Program menu as the top level for every menu selection. Each time you prepare the INTERVIEW for the communications environment, you will start with the Program menu and move down into the other menus.

All setup, trigger, and programming menus are accessible from this main menu. In general, use the PROGRAM key from other Program–mode screens or from Run mode to access this menu.

> **NOTE:** There are two exceptions to this rule about the PROGRAM key. First, if the file named *user_intrf* is located in the */usr* directory of the hard disk during power–up, pressing PROGRAM loads, compiles, and runs the program in *user_intrf*. See Section 2.2(A). Secondly, if you select and run a program from an Easy View menu, pressing PROGRAM from Run mode returns you to the Easy View menu instead of the main Program menu. See Section 4 on Easy View operation.

(B) Configuring Menus

You may configure menus in any order you wish; however, we suggest you configure the Layer Setup screen before programming on the Protocol Spreadsheet or the Display Setup screen, since the selections available to you are governed by the protocols loaded on the Layer Setup screen.

All options on a particular menu are summarized as a diagram which appears in this manual at the beginning of the section which covers that menu. Programming options available in each protocol package are discussed in a section pertinent to the protocol and layer.

To select a submenu from the main Program menu, use ROLL FWD or ROLL BACK key, or press the labeled softkey. Enter the menu by pressing the highlighted function key or RETURN. The labels on the screen guide you through each step. For example:

- Access the Layer Setup screen from the main Program menu by pressing F5.
- Access the Disk Maintenance screen from the main Program menu by pressing F7 for utilities, then F3 for Disk Maintenance.

Other uses of the function keys are described in Section 3 and in sections pertinent to the various submenus.

Once you have reached the menu you want, the function keys lead you through selections, down to the smallest level of detail. You may also use cursor keys ↑ and ↓ to move up and down through the menu fields. ← and → move the cursor across a menu from field to field.

If you wish to return to the last screen, press PREV PAGE. Should you wish to review the menus you have configured, this key takes you back one screen at a time until you reach the main Program menu. Of course, you may press PROGRAM to start again from the top of the menus. When you are ready, press RUN to start a testing session, no matter what menu is displayed.

Read Section 3 for a further discussion of these and other keys.

(C) The Setup Menus

Some menus in the setup group (listed at the top of the Program menu) are used to specify how data is sent, received, recorded, and displayed. If you have an ATM unit with the ATM hardware (an ACE board and an AIM board in the top enclosure, the menu allows setting parameters for sending and receiving Asynchronous Transfer Mode (ATM) data, along with the choice of disabling the WAN protocol data functions. Sample setup menu softkey selections are shown in Figure 2-4. Note that BERT (F5) will appear or, if you have an ATM unit, an AIM Interface (F6) will appear, but these two selections will not appear simultaneously.

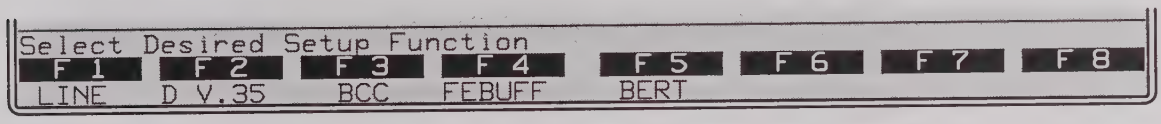


Figure 2-4 Sample softkey rack on Setup Menu screen for *TURBO units*: installed Test Interface Module is Dual–Port V.35. [Note that *either* BERT (F5) or an AIM Interface (F6) will appear (see Figure 2-5), but not both.]

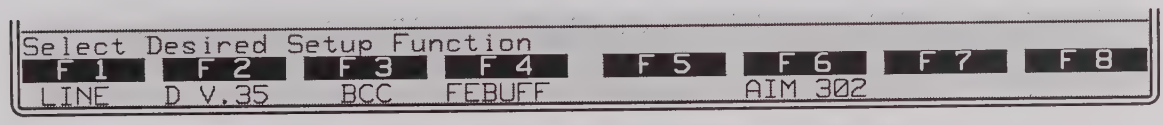


Figure 2-5 Sample softkey rack on Setup Menu screen for *PLUS* ATM units: installed Test Interface Module is Dual–Port V.35 and installed ATM Interface Module is AIM 302. [Note that BERT (F5) is neither applicable or accessible.)

1. *Line Setup.* The Line Setup screen allows you to designate the role which the INTERVIEW is to play in testing—whether it is to monitor passively or participate in an active dialogue as DTE or DCE. Menu settings also determine the source of the data and the data clock as well as the characteristics of the data stream you expect to receive or send. These characteristics include the scheme for character encoding/decoding and the format in which blocks of data are sent and received (i.e., synchronous, asynchronous, bit–oriented, or isochronous). The Line Setup screen is described in Section 5.

2. *Display Setup.* The Display Setup (on the Line Setup screen) provides alternative types of display to aid in analysis. On this screen, designate how you want data to appear. (Data may be displayed as a stream of bytes alone or in conjunction with lead transitions; summarized in a protocol trace or customized trace format; or tracked on one of two statistical displays.) Then, as you require different types of analysis, use function keys to change from one type of display to another while testing, without returning to the menu. The Display Setup screen and the different types of displays are described in Section 6.

3. *Record Setup.* The Record Setup (also on the Line Setup screen) defines recording conditions for data acquisition tracks on disk; or for RAM (RAM

capacity is 256 Kbytes in the 8100 *TURBO* and 8200 *TURBO*; 1 Mbyte in the 8600 *TURBO,* 8700 *TURBO* and 8800 *TURBO*). This screen does not influence the data stored in the character buffer. Storage of data in the screen's character buffer can be controlled from the Protocol Spreadsheet using the Capture command. The Record Setup screen is described in Section 7.

4. *FEB Setup.* Data and control–lead signals entering the INTERVIEW or generated internally are routed from the receivers through a front–end buffer (FEB) before being presented to the screen and to the trigger program. Data bits are buffered automatically in the FEB. The buffering of other events—control leads, idle bits, time ticks, and frame timestamps—can be enabled or disabled on the Front–End Buffer Setup menu, explained in Section 9. Note that this selection line along with its function key label (see Figure 2-4) will not appear on the Setup Menu when DISABLE is selected on the Line Setup screen—see Section 5.1.

5. *Interface Control Setup.* The Interface Control Setup screen provides programming selections for Test Interface Modules you have installed. The menu may differ slightly according to whether EMULATE DTE or EMULATE DCE is the **Mode** selection on the Line Setup Screen. The menu is specific for the installed TIM (see Section 12). Note that this selection line along with its function key label (see Figure 2-4) will not appear on the Setup Menu when DISABLE is selected on the Line Setup screen—see Section 5.1.

6. *BCC Setup.* The BCC Setup Menu controls and displays the values of the INTERVIEW's block–check parameters. For more information on block checking, refer to Section 10. Note that this selection line along with its function key label (see Figure 2-4) will not appear on the Setup Menu when DISABLE is selected on the Line Setup screen—see Section 5.1.

7. *BERT Setup.* The INTERVIEW can transmit and analyze Bit Error Rate Tests (BERT). Once you have selected the BERT mode on the Line Setup menu, select appropriate parameters on the BERT Setup menu. Note that this selection line along with its function key label (see Figure 2-4) will *only* appear on the Setup Menu when BERTDCE or BERTDTE is selected on the Liine Setup screen. See Section 5.1 and Section 11.

8. *AIM Interface Setup.* The AIM Interface Setup screen provides programming selections for the ATM Interface Module (AIM) you have installed. The menus may differ slightly according to your selections. See the ATM technical manual. This selection will *not* appear if **Mode:** AUTOMONITOR , BERTDCE , or BERTDTE is selected on the Line Setup screen.

(D) The Trigger Setup Menus

The next three groups of menus, Triggers, Spreadsheet, and Statistics, are programming menus which you use to establish interactive dialogues, create test scenarios, and make and display measurements.

NOTE: BERT testing is handled separately, on the BERT Setup screen. See Section 11.

The 16 identical Trigger Setup screens are a limited set of test conditions and actions grouped in a standard menu format. The set of conditions offered on these screens is described in Section 25. Trigger Setup actions are described in Section 26. The Protocol Spreadsheet provides a wider range of conditions and actions which vary, according to the layer and the protocol you are programming. Conditions and actions available on the spreadsheet are covered in Sections 31 and 32 and in sections dedicated to each layer protocol.

Trigger Setup screens and the Protocol Spreadsheet may be used together as described in Section 23. Counters and timers of the same name may be shared between the two, as can the flag bits from the Trigger Setup screens, which are accessed as flags on the spreadsheet under the name **trig_flags**.

(E) The Protocol Spreadsheet

The Protocol Spreadsheet, a more flexible programming tool with more options than the Trigger Setup screens, is initially a blank menu. Legal programming options are presented as function key labels at the bottom of the screen. Create your program by pressing the necessary function keys. Your entries will be posted on the screen. As you make entries, the function keys reflect the new options enabled.

You may also type your program directly onto the screen, as long as you observe syntax and use exact keywords (as they are posted on the screen; *not* as they are abbreviated in function key labels.) Programming options and errors will still be tracked.

Syntax errors are indicated by strike–throughs when you have completed an entry. (If you are typing onto the spreadsheet, completing an entry usually means pressing the space bar, pushing DONE or RETURN, or moving the cursor to a different location.)

1. *The spreadsheet pattern.* The Protocol Spreadsheet expects a certain pattern of entries. To gain access to the set of trigger conditions and actions at each layer, you must first identify what layer you are programming, then what test you are developing, and finally the name of the state which will contain the triggers you create. Once you have named the state, press the function key for CONDITIONS:. At this point, actual programming options will appear. When you complete the Conditions portion of a trigger, press DONE and then press the function key for ACTIONS: to display possible trigger actions. Use the NEXT_STATE: action to indicate movement to another state. The NEXT_STATE: action must be followed by a state name. It can move to any state within the test; the NEXT_STATE: NEXT action moves the test to the following state on the spreadsheet. Programming concepts related to these selections are described in Sections 23, 24, and 28.

Here is an example of the spreadsheet pattern you will see repeatedly:

```
LAYER: 1
  TEST: example
    STATE: begin
      CONDITIONS: One or more conditions appear here.
      ACTIONS: One or more actions appear here.
       NEXT_STATE: second

      (NEXT_STATE will not follow every trigger.  Also, it may replace
ACTIONS)
```

2. *Additional spreadsheet capabilities.* Constants may be used on the spreadsheet to represent repeated values or text, and they may be placed so that they apply to all or part of a program. Constants are explained in Section 29. C programming language can be introduced at any location on the spreadsheet to create new testing conditions and actions and generally increase program flexibility. C is introduced in Section 55.

(F) The Statistics Screens

The INTERVIEW has two different statistics menus, one in tabular form, the other in bar−graph format. Both are accessible by softkey while you are analyzing data. The value of counters and timers named in triggers can be tracked in statistical screens, once their names have been entered on the corresponding menus. Current, last, minimum, maximum, and average values are tabulated. Values for several counters and timers may be totaled by an accumulator. Bar graphs can be scaled and color−keyed. Refer to Sections 21 and 22 for information on statistics menus and displays.

(G) The Layer Setup Screen

Before you program the Protocol Spreadsheet, you are advised to load the protocols you intend to use. Protocols are selected and loaded from the Layer Setup screen. Your selections, once loaded, determine the set of program conditions and actions which appear on the spreadsheet.

With most protocols, a secondary screen loaded with the protocol allows you to modify common parameters for the protocol.

For more information on the Layer Setup screen, see Section 8.

(H) The File Maintenance Screen

The File Maintenance screen is the user's interface with the filing system. The menu facilitates saving and loading programs, renaming or deleting files. It allows you to consult the contents of any directory, create a new directory, and write−enable or write−protect a file. From this menu, you may structure your own filing system according to your needs.

NOTE: The files pertaining to the operating system and menu selections are stored primarily in the */sys* directory, with some files stored in the */usr* directory. These files should not be deleted or moved. You are otherwise free to manipulate the filing system as you wish.

Section 15 explains file and directory pathnames, how to set up a file hierarchy, how to move through the filing system, and how to use the various maintenance commands.

(I) The Utilities Menus

The Utilities menus are used to manage peripherals: disk, printer, the internal time–of–day clock, and color monitor.

The Disk Maintenance menu allows you to allocate disk space for data and programs, to transfer data from one storage medium to another, and to duplicate the contents of one disk onto another. Section 14 describes disk maintenance commands and their use in detail.

The Printer Setup screen allows you to configure the INTERVIEW to control most serial ASCII printers. The various menu fields are described in Section 16.

Consult the time or modify it on the Time/Date Setup screen. The time and date which appear here are used in time–stamping data blocks and user files. See Section 17 for details on this screen.

The Miscellaneous Utilities screen provides mapping of black and white to color enhancements for external monitors. Once the mapping is completed, these enhancements may be placed under trigger control for the production of highlighted data. Refer to Section 18 for further information on this screen.

The Easy View Setup screen governs the operation of the Easy View system. Use the menu selections to: enable Easy View, automatically enter Easy View after power–up, keep Easy View menu information in memory, and display a warning message before running programs from Easy View. See Section 19.

(J) Remote Control Setup Menu

The Remote Control Setup Menu of the INTERVIEW unit is shown in Figure 2-6. (This menu can be viewed on the PC as well when the remote connection is completed.) After your information is added to the fields, press [XEQ] to register the changes and [SHIFT]–[SAVE] to save the setup parameters to disk. (The IP addresses in the *Host Address*, *Subnet Mask*, and *Gateway Address* fields will become effective the next time you boot up the INTERVIEW.)

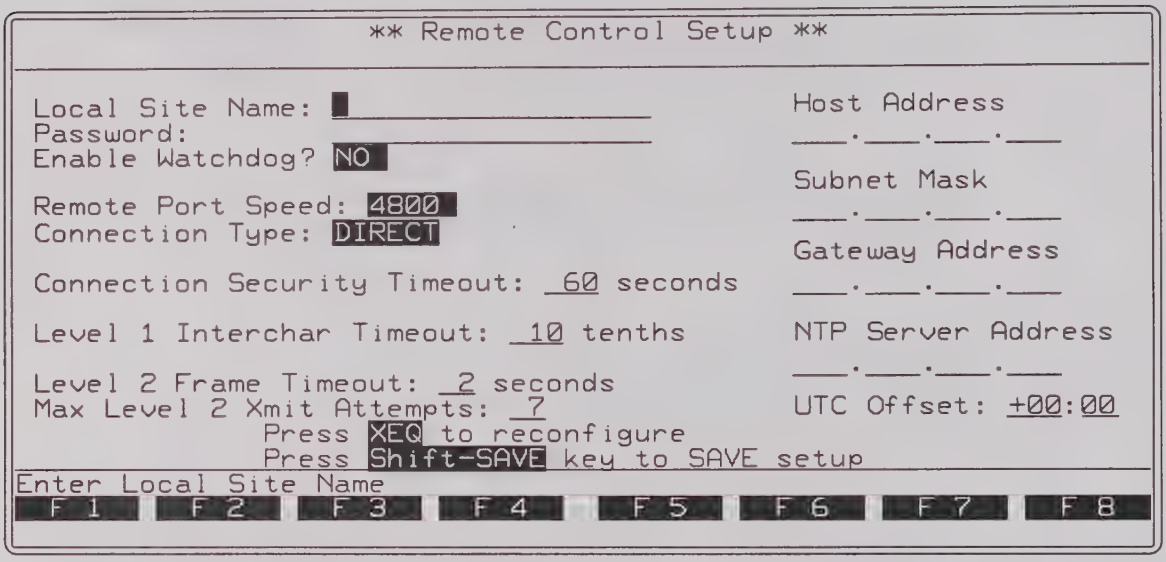


Figure 2-6 Remote Control Setup Menu on the INTERVIEW

NOTE: If a remote session is in progress, these parameter changes will not be effective until after the current remote session has been completed.

1. *Local Site Name*. Identify the local site for the INTERVIEW in the **Local Site Name** field using up to 18 ASCII characters. This entry will become the site name displayed on the controlling PC's screen for the INTERVIEW after a connection has been established. If this field is left blank, the PC will recognize the INTERVIEW as *No Name*.

2. *Password*. If you wish to password–protect the remote INTERVIEW, enter a password of up to 18 ASCII characters in this field. If no password is entered in this field, any password submitted by the PC will be accepted.

   NOTE: When an entry is made in this field, the Password you enter in the Manual Connection dialog box on the PC must be an ***exact case–sensitive*** match in order to achieve a PC–to–INTERVIEW connection.

3. *Enable Watchdog*. The INTERVIEW is equipped with a "watchdog timer" feature which, when enabled, will automatically reboot the unit in the unlikely event of software failure. This automatic reboot feature replaces the requirement for someone to travel to the remote site to manually reboot the unit, saving both time and effort. This can be enabled either in hardware or via the software.

   The GBM board is jumpered when it leaves the factory to disable the watchdog timer. The **Enable Watchdog?** query allows you to enable the watchdog timer (with

YES) or leave it disabled (the default NO). A YES entry will be valid for the current session once XEQ is pressed; to change the default to YES , press SHIFT — SAVE .

Once enabled, the watchdog timer circuit can be disabled again in the following manner: select **Enable Watchdog?** NO , press SHIFT — SAVE to save the selection, and press CTRL — SHIFT — DELETE CHAR or CTRL — SHIFT — DELETE LINE to reboot the system; this prevents accidental disabling of the feature.

> **NOTE:** The **Enable Watchdog?** field is available only if the INTERVIEW is equipped with system software version 9.02 or higher (7.02 or higher for 6000 Series units) *and* revision 1 or higher of the GBM board. If the unit has a lower revision of either the GBM board or the system software, the **Enable Watchdog?** selection will not appear on the Remote Control Setup screen. For these units, the timer is enabled through moving a jumper on the GBM board, as described in the Appendix H5.

4. *Remote Port Speed*. The baud rate of the INTERVIEW's remote port hardware is selected in the **Remote Port Speed** field. Four choices are available: 2400 , 4800 , 9600 , and 19200 bits per second.

5. *Connection Type*. Choose the type of connection on the remote port via the **Connection Type** field. The selections are DIRECT for a hard—wire connection and MODEM for connection via modem. When MODEM is selected, an additional field appears: **Modem Script**.

6. *Modem Script*. The **Modem Script** field defines the script file that will be run during the connect and disconnect sequences. The default selections provided with the operating system software are HAYES , UDS_V32 , and INT_V42 . You may also opt to create a selection for another type of modem by copying an existing ASCII file and editing it. These ASCII files are found in the *HRD/sys/rc* directory on the INTERVIEW and may be edited via the Spreadsheet Editor (see Section 2.6).

7. *Connection Security Timeout*. This field applies to incoming calls to the INTERVIEW *only*. The **Connection Security Timeout** is the number of seconds before the unit hangs up the modem if a connection is not established. If this timer expires before a valid link establishment request is received from the PC, then the physical link will be dropped. (This includes hanging up the phone for modem links.) Enter any value from 0 to 999 seconds; the default is 60 seconds.

The next three parameters are used to determine how the Remote Control Protocol will respond to data loss or error. The first two are user—defined timeouts and the third sets the maximum retransmit count at Level 2. Table 2-1 serves as a summary of the Level 1 and Level 2 timers and the maximum number of retransmissions for specific types of connections; it is meant for use as a guide only.

8. *Level 1 Interchar Timeout*. The time to expiration of the intercharacter timer is defined in the **Level 1 Interchar Timeout** field. Enter any value from 1 to 999 tenths of a second; the default is 10 tenths (*i.e.*, 1 second).

This timer, used to detect the loss of inbound asynchronous characters due to transmission errors before a frame is complete, determines the maximum allowable time between two consecutive received characters. It is enabled after the first character of each potential frame is received, reset upon receipt of each character, and disabled after receipt of the final character in a frame. If the timer expires before the next character of a frame arrives, all characters received are *flushed* from the frame buffer. Generally, this will result in a Level 2 Frame Timeout and the frame will be retransmitted.

9. *Level 2 Frame Timeout*. The time to expiration of the Level 2 timer is defined in the **Level 2 Frame Timeout** field. Enter a value from 1 to 999 seconds; the default for this parameter is 2 seconds.

   The Level 2 timer determines the maximum allowable time between the transmission of a frame and the receipt of an acknowledgment from the far end. It is enabled after the last character of each frame is sent and is reset upon receipt of an acknowledgment. If the timer expires before the acknowledgment arrives, the frame will be retransmitted (up to the number of times specified in the **Maximum Level 2 Xmit** field).

10. *Max Level 2 Xmit Attempts*. The value in the **Max Level 2 Xmit Attempts** field sets the maximum number of times that a frame may be transmitted without receiving an acknowledgment before a protocol error is registered. If this count is exceeded, the communication link is disconnected. Generally speaking, the noisier or more error–prone the link, the larger this number should be. Enter a value in the range of 1–99; the default is 7.

Table 2-1
Summary Guide of Level 1/Level 2 Timers and Maximum Number of Retransmissions for Specific Connections

Connection Type	Level 1 Timer (tenths of sec)	Level 2 Timer (secs)	Maximum Retransmissions
High–Speed Direct	5	1	7
Low–Speed Direct	10	2	7
High–Speed Modem	5	1	7
Low–Speed Modem	10	2	7
Packet–Switched Networks	20	5	3

The final five fields on this screen, located on the right–hand side of the display, allow you to configure the INTERVIEW for use on an Ethernet network; these fields will be displayed only if your unit is Ethernet–ready. The first four fields each consist of an IP address displayed as four 3–character entry fields with periods between them. The default condition for these fields is blank.

11. *Host Address*. Enter the designated IP host address for the INTERVIEW unit in this field.

12. *Subnet Mask*. Enter the subnet mask used for your network in this field.

13. *Gateway Address*. Enter the gateway address for your network in this field.

14. *NTP Server Address*. Enter the IP address for the Network Time Protocol server with which the INTERVIEW unit is to synchronize in this field. Set this address to all zeroes when no synchronization is desired.

> **NOTE:** *Host Address*, *Subnet Mask*, *Gateway Address*, and *NTP Server Address* are specific to the network and NTP server to which you are connecting; contact the network administrator for this information. Be sure to enter addresses in these fields accurately, as incorrect/ invalid IP addresses may affect the network adversely. *You must reboot the INTERVIEW before the Host Address, Subnet Mask, and Gateway Address changes go into effect.*

The last field is used in conjunction with the Network Time Protocol[1]:

15. *UTC Offset.* Enter the difference between your local time and Coordinated Universal Time (UTC) in this field. The value should be in + or − hours (and minutes, if necessary), *e.g.,* −05:00 for Eastern Standard Time, −04:00 for Eastern Daylight Time in the United States.

2.6 Customizing the Modem Script Field Selections

As most intelligent asynchronous modems require the proper setup to function, you may need to make a new modem selection for the specific model you are using. You can do this by creating a new script file in the *HRD/sys/rc* directory of the INTERVIEW.

> **NOTE:** This program requires that the modem be set up to pass through the X–ON/X–OFF characters. Be sure to address this in your modem script.

To create a customized **Modem Script** menu selection called XXXX, do the following:

- Copy the modem ASCII file for the HAYES selection to another file, *XXXX*:
  - — In Program mode, go to File Maintenance via PROGRAM, **FMAINT**
  - — Press **CHNGDIR**, select **HRD**, type in */sys/rc*, and press XEQ
  - — Press **COPY**
  - — Press the softkey for the disk in which the file is residing, **HRD**
  - — Type in */sys/rc/HAYES* and press RETURN
  - — Press the softkey for the disk in which the new file will be residing, **HRD**
  - — Type in the name of the new file */sys/rc/XXXX* and press XEQ

  The new ASCII file, named *XXXX*, is ready for editing.

[1] Portions of the software for INTERVIEW support of NTP are Copyright © David L. Mills 1992, 1993, 1994, 1995, 1996.

- Edit the newly created file *XXXX* in the Spreadsheet Editor:
  - Press [PROGRAM], **SPDSHT** to access the spreadsheet
  - Read the file into the spreadsheet buffer: press [EDIT], **BLOCK, IN/OUT, READ/U**
  - Type in *HRD/sys/rc/XXXX* and press [XEQ]
  - Clear the inverse text by pressing [DONE], **CLEAR**
  - Edit the file to fit your modem's requirements using the ASCII keyboard and arrow keys

- Copy the edited file from the Spreadsheet Editor to *HRD/sys/rc/XXXX*:
  - Block the entire new text: scroll to the beginning of the text and press **BEGIN**, then scroll to the end of the text and press **END**
  - Write it back to the disk by pressing **IN/OUT, WRITE/U**
  - Type in the name of the file, *HRD/sys/rc/XXXX,* and press [XEQ]
  - Press [PROGRAM] to exit the Spreadsheet Editor

Reboot the unit to activate the new selection "XXXX" for the **Modem Script** field.

2.7 Running a Test Program

Press [RUN] to compile and execute a test program. As the program is compiled, a message appears at the top of the screen to indicate the phase that the compiler is in. The longer and more complex the program, the greater the compile time that is required. The preparations being made in each phase are briefly outlined below.

(A) Test Preparation

1. *Phase 1.* Trigger Setup screens are converted to the Protocol Spreadsheet format, the Protocol Spreadsheet is converted to C, and the C Preprocessor directives are acted upon.

   NOTE: Compilation time is somewhat faster if all triggers are programmed directly on the Protocol Spreadsheet.

2. *Phase 2.* The Program is compiled.

3. *Phase 3.* The number of processors and the configuration of the equipment (into which the program is to be loaded) are determined, and internal packages to support the user's program are structured. Also during this phase, linkable–object files referenced on the Protocol Spreadsheet are found and the compiled code in them combined with the compiled spreadsheet program.

4. *Phase 4.* Resources are allocated for each separate task in the program.

5. *Phase 5.* The run–time operating system is generated.

6. *Phase 6.* All code is linked.

7. *Phase 7.* A memory image of the code which can be run by the processors is built.

(B) Rerun Without Recompiling

After a program has compiled once, it will enter Run mode "immediately" (within two seconds) after RUN is pressed in subsequent executions of the program—assuming that no substantive changes have been made in the program in between the two runs. Substantive changes may be defined as those changes made to menus and fields not listed in Table 2-2. That is, changes to fields covered in Table 2-2 will not necessitate a recompile and will not prevent the unit from entering Run mode immediately.

Any changes to the Trigger Setup menus or to the Protocol Spreadsheet will be considered substantive and will necessitate a new compile.

Changes to the tabular or graphic statistics screens will not cause the program to recompile. However, if a counter or timer is added to the screen, it will not update until the program is recompiled. You can force a recompile by holding down CTRL and pressing RUN.

Table 2-2
Fields That Can Be Changed Without Causing Recompile

Menu	Field
Line Setup	Disk No (recorded data must be same *type*) Block No Clock Source Speed NRZI MIL
Display Setup	Display Mode Type RTS/CTS? CD? DTR/DSR? C/I? (for X.21 Test Interface Module) Layer (Protocol or User Trace; not Program) Statistics Type
Record Setup	Disk No DAT Record Size Initial Cond Stop At
Front–End Buffer Setup	Idle Suppress DTE DCE
T1/G.703 Transmit Setup	(all fields)
T1/G.703 Data Path Setup	(all fields)
T1/G.703 Line Setup	(all fields)
Tabular Statistics	(all fields)
Graphical Statistics	(all fields)
File Maintenance	(all functions except LOAD)
Date/Time Setup	(all functions)
Printer Setup	(all fields except Redirect Run Mode Output)
Disk Maintenance	(all functions)
Misc Utilities	(all fields)
Easy View Setup	(all fields)
Remote Control Setup	(all fields)
AIM Interface Setup	(all fields)

(C) Errors Detected When A Program Is Compiled

Trigger programs and spreadsheet programs with syntax errors or other errors will not compile successfully and will prevent the unit from executing the test program.

If an error is detected during the compiling phases (after you have pressed [RUN]), the user is returned automatically to the Protocol Spreadsheet unless the program has been started via the Easy View menu (in which case, you will be returned to Easy View).

While a program is being compiled, errors are flagged and diagnostic information is stored. This information is made available to the user when the unit is returned to Program mode. A diagnostic message for the first error found is automatically displayed at the top (second line) of the Protocol Spreadsheet screen. The cursor is automatically moved to the error.

Press GO–ERR once more to move to the next error. For each error, a diagnostic message is displayed. The search for errors stops at the end of the file and the message "No more errors" is displayed.

GO–ERR also calls up diagnostic information on trigger–menu errors.

Error messages are listed in Appendix A of this manual.

(D) Recoverable Errors During Run Mode

The following messages indicate MPM processor errors:

MPM –– Processor Fault
MPM –– Divide Fault
MPM –– Bus Error
MPM –– Stack Fault
MPM –– Memory Fault

Do not turn off the INTERVIEW when any one of these errors is displayed in Run mode. Instead, press [PROGRAM] and check your program, since these messages indicate programming problems that cannot be displayed as syntax errors but which do prevent your test from running properly. Consult Appendix A1 for an explanation of these messages. Once you have revised the test, try running it again. If you cannot resolve the problem, save a copy of the program and contact Customer Service/Technical Support.

2.8 Data Flow

Figure 2-7 diagrams the movement of data between the various functional components of the INTERVIEW. The diagram provides "clues" to many of the operating characteristics of the unit. For example:

- The front–end buffer (FEB) lies squarely in between the line interface and (1) the recording medium and (2) the program logic. This means that control leads may or may not be recorded and may or may not be seen by the triggers—depending on the FEB setup (Section 9).
- Line data may be recorded directly to disk as bit–image data, recorded to bit–image RAM and then transferred to disk, or captured as character data in the display buffer and transferred to disk.
- Once control leads and time ticks or frame timestamps (that is, the original timing values) are recorded alongside character data, they are locked in. Since the FEB is not on the playback path for character data, FEB selections do not apply.
- Bit–image data, however, does pass through the FEB during playback. Except for the **Idle Suppress** and **Frame Timestamps** fields, FEB selections apply. This means that control leads and time ticks, if recorded with the data, *must* be enabled in order for the program logic to detect them. (For playback of bit–image data, the **NRZI** selection on the Line Setup menu also does not apply.)
- Only data on disk may be played back.
- Front–panel green–red LED's are not disabled for line data, and do not blink for recorded data.
- Not only characters but also leads and time ticks (or frame timestamps), if enabled in the FEB setup, are captured automatically in the display buffer (that is, the screen buffer or character RAM).

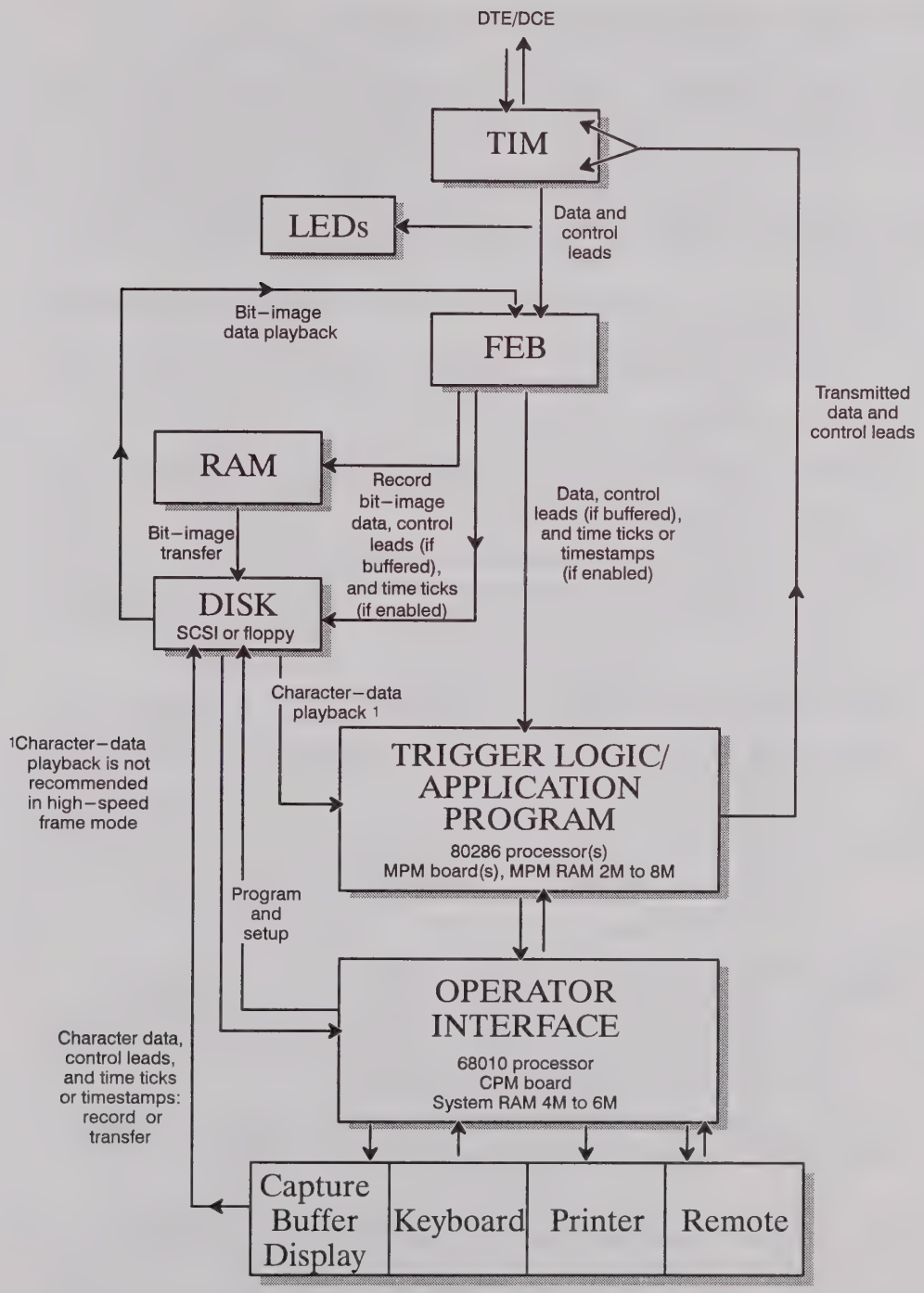


Figure 2-7 INTERVIEW 8000 Series functional diagram.

2.9 How to Correct Common Problems

(A) Unit enters Run mode even though I press the Program key.

A file called *user_intrf* was located in the */usr* directory of the hard disk during boot–up. In this situation, the program in *user_intrf* is automatically loaded, compiled, and run each time the PROGRAM key is pressed. Press SHIFT – PROGRAM to enter Program mode, or rename *user_intrf* and power–up again. See Section 2.2(A).

(B) Unit does not execute Run: Protocol Spreadsheet returns to screen instead.

An error was detected during the compiling phases (after you pressed RUN). See Section 2.7(C).

(C) My program does not run; and instead, I get a message about an "unresolved reference."

Your program is asking the compiler for a send routine that is not available in Monitor mode. Switch to an Emulate mode, or modify the program. For programs with C coding, this message usually means that a routine has not been declared or defined.

(D) Protocol Spreadsheet program which was just loaded shows syntax error strike–throughs which weren't there before.

Missing softkey selections and pervasive strike–through's on the Protocol Spreadsheet indicate that the correct Layer Personality Package has not been loaded. To correct the problem, return to the Layer Setup screen, insert disks if necessary, and check Personality Package and drive selections. Then press XEQ to load the packages manually.

(E) EIA trigger condition does not come true, even though the front–panel LED indicates a status that makes the condition true. For example, an EIA RTS ON condition is not coming true, even though the RTS LED is bright red.

EIA status is not detected by the triggers if **Buffer Control Leads:** NO is the selection on the Front–End Buffer Setup menu. See Section 9.

If the data is being played back from disk and the FEB Setup menu was not configured to buffer control leads at the time the data was recorded, the leads are no longer available for triggering.

Front–panel LEDs *always* reflect line status, *never* the status of recorded leads.

Note also that an EIA condition that is the only condition on a trigger menu (or the only condition associated with an action or set of actions on the Protocol Spreadsheet) is *transitional*. It is only true when it changes to true. To check the current *status* of an EIA lead regardless of transitions, pair the EIA condition with a don't–care character condition (see Section 25.4) or with an ENTER_STATE condition on the Protocol Spreadsheet. See Section 31.3(A).

(F) Lead lines on the data–plus–leads display are not transitioning, even though the front–panel LEDs for the same control leads are blinking.

The leads on the data–plus–leads display are also enabled/disabled by the Buffer Control Leads field on the FEB Setup menu. See (E), directly above.

(G) My "current" timers seem to be incrementing on the Tabular Statistics screen, but the other statistical columns always show 0, even when I take statistical samples.

The "current" column is derived from a millisecond "clock on the wall," while the statistical values may be calculated on the basis of time ticks that occur at one–second intervals. Your timer samples may be less than 0.5 seconds, in which case, they are rounded to zero.

Check the FEB Setup menu, and solve the problem in either of two ways. Turn Time Ticks off, and the statistical columns will use the wall clock. Or change the Tick Rate from one second to ten milliseconds or smaller (down to ten microseconds).

(H) I am trying to send a transmit string from a trigger menu (or from Layer 1), but my data does not appear on the screen.

When you are having trouble transmitting, always go to the Line Setup menu and look at the **Clock Source** field first. You must have clock to transmit, whether internal or supplied by an external DCE.

Another frequent problem is receiver synchronization. When you try to transmit, does the front–panel LED for your transmit lead (TD or RD) blink rapidly? If it does, then you are transmitting successfully, but your receivers may not be synchronizing with the data.

Check the **Sync Char** field on the Line Setup menu. Also be sure that the sync pattern is part of your transmit string. You must supply these characters yourself.

(I) I have loaded in X.25 packages at Layer 2 and Layer 3, and I am trying to send a data packet from Layer 3; but my data does not appear on the screen.

The INTERVIEW is a *layered* emulator. The significance of this is that Layer 3 and higher layers (in Emulate modes) have no direct access to the physical layer, Layer 1. In practice this means that a RCV condition at Layer 3 *does not see packets on the line*. It only sees packets that are delivered up from Layer 2 by a *user program* at that layer.

Similarly, a SEND action at Layer 3 does not in itself send a packet out onto the line. A SEND action merely delivers the packet to Layer 2—provided that Layer 2 has indicated its readiness to receive data from above.

The following program is not any sort of complete Layer 2 emulation. It is the *minimum* program that must be entered at spreadsheet Layer 2 in order for a Layer 3 program to have access to the data line. Once this Layer 2 program is entered, Layer 3 can send packets out onto the line and receive packets from the line.

```
LAYER: 2
   STATE: datalink
      CONDITIONS: DL_CONNECT REQ
      ACTIONS: DL_CONNECT CONF
      CONDITIONS: DL_DATA REQ
      ACTIONS: SEND INFO " ((DL_DATA)) "
      CONDITIONS: RCV INFO
      ACTIONS: GIVE_DATA
```

The elements of this program are discussed in Section 34, OSI Primitives on the Protocol Spreadsheet, and in the programming example in Section 37.9.

(J) I'm trying to load a program; but instead, I get an error message about loading a layer package.

When a program or object file is loaded from the File Maintenance screen, the system must be able to locate any layer personality package that the program requires. If the necessary layer package cannot be found, the file will not be loaded.

> **NOTE:** Optional applications programs are available for the INTERVIEW 8000 Series. Make sure that necessary layer packages are accessible when you load these programs.

In the INTERVIEW 8000 Series, all layer packages are accessible at all times since they reside in the */usr/layer_pkgs* directory on the hard disk. See Section 8.

(K) My INTERVIEW is acting differently than normal.

Collected dust and dirt on the fan filter may be blocking the flow of air into the unit. It may be overheating. Follow the instructions in Section 1.5(D) on cleaning the fan filter. If the problem persists, contact Customer Service/Technical Support.

(L) I power up my INTERVIEW and it gives a message about not being able to find *mstrmenu.cmp*.

You have a *TURBO* unit that is trying to load the menu information for the Easy View system. First, load *HRD/AR_APPS/UTILS/EZVUmaint* onto the hard disk. Next, compile *mstrmenu.txt* to try to reinstall the file: press any key except PROGRAM or RETURN, and use the COMPILE softkey. This should reinstall the file.

If the file is not reinstalled, and if you have the Easy View disks, duplicate their contents onto your hard disk. (See Section 14 for information on the Duplicate Disk command.) If you do not have the Easy View disks, go to the Easy View Setup screen, select **Enter Easy View After Power-Up:** NO , and save the setup. (See Section 19.)

(M) I power up my INTERVIEW and it gives a warning message about the CPM.

During power-up, *TURBO* units check the revision levels of all boards for compatibility. Even if the revision levels are not as expected, the general operation of the unit is unaffected. Contact Customer Service for further information.

(N) I correctly set up my INTERVIEW menus for emulation, but nothing happens when I try to begin emulating.

Many of the later Test Interface Modules have a switch on the TIM which allows you to enable or disable emulation. Check to see that this switch is in the ENABLE position for emulating. For further information, see the TIM documentation.

(O) When running an application program at T1 rates, I get an "FEB Overflow" message.

Early application programs (especially those developed for some of the older protocols like SNA) were designed for monitoring specific interface activity such as leads and time ticks which are important when running at 9600 bps. These programs, some of which are resident in Easy View, will not run at today's high speeds specifically because of their excess "baggage" (overhead) which slows down the program. To run such programs, turn off the leads, change from time ticks to frame timestamps, select bit-image recording on the Line Setup screen, and run the program in High Speed Frame Mode. This should allow the program to run at T1 rates.

(P) I'm a new INTERVIEW operator; where do I begin?

Your first step should be to read the *30 Minutes to Programming the INTERVIEW® 8000 TURBO Series* (951-B0431-01) included with your unit. Work through the exercises with the data disk provided to familiarize yourself with the unit's operation. Secondly, an excellent source of general information is found on-line: go to the Easy View Main Menu by pressing EZ VU (if the menu isn't already displayed) and press ? for Help on-line. A third source (also on-line) is located on a submenu of the Easy View Main Menu; arrow down to and select the OPERATION menu. Selecting any topic from this submenu gives detailed information on that subject directly on the screen. And, of course, the Table of Contents and Index in this technical manual also direct you to specific topics.

(Q) My unit is running at a very slow rate.

It is a simple fact that the more tasks there are to accomplish, the longer it takes to complete them—time is a relative factor. This is also true of the INTERVIEW. When line rates are relatively high, it may be best to strip your program of extraneous

tasks which would slow down the unit. This may include loaded Layer Personality Packages that are not being utilized by your application, recording data with tick rates, *etc.* Refer to the *INTERVIEW Technical Manual* for more information on performance optimization.

If you are operating an INTERVIEW unit other than an INTERVIEW 8800, you may also wish to consider upgrading your unit to the next higher INTERVIEW (e.g., 8200 to 8600), as the upgrade will add more processing power. (This is true for all upgrades except for the INTERVIEW 8100 to INTERVIEW 8200 upgrade.)

(R) I'm trying to run an application program that has worked before, but it won't compile or run.

If you are operating a non–*TURBO* unit with software revision 8.00 (or higher) installed, there may not be enough system memory available to run large application programs. You may need to upgrade your system.

Also, an application specifically designed for operation on an INTERVIEW 8000 Series unit may not compile on a 7000 Series unit due to memory capacity constraints.

(S) I'm in the Easy View menu system, and I want to run an application with a dual–port V.35/EIA–530(449)/EIA–232 TIM installed to test at the EIA–232 interface. What should I do?

To enable the correct parameters for your interface, you must configure the V.35 Interface Setup screen *before* you load the application. This is done in the operating system; to access the operating system menus from the Easy View menu system, press EZ VU. This will display the Program Menu.

From the Program Menu, press F1 SETUP, then press F1 LINE, and configure the Line Setup screen for the data you wish to test.

Return to the Program Menu by pressing PROGRAM. Once again press F1 SETUP, then press F2 D V.35, to access the V.35/530/232 Dual–Port Interface Control screen.

Choose SINGLE PORT or DUAL PORT for the Mode field. If you have selected SINGLE PORT for this field, move to the Connector field. Choose V.35, EIA–530(449), or EIA–232 (your indicated choice) and configure the remainder of the screen as appropriate for your link.

To run an application program from Easy View, then press EZ VU and access the program to run.

Refer to Section 5 for information on the Line Setup screen and to the V.35 Multi–TIM documentation which accompanied your dual–port V.35/EIA–530(449)/EIA–232 TIM.

2.10 Optimizing Recording Rates

(A) Recording with Standard Hard Disk Drive (80 Mbyte)

It is a simple fact that the more tasks there are to accomplish, the longer it takes to complete them—time is a relative factor. This is also true of the INTERVIEW. When line rates are relatively high, it may be best to strip your program of extraneous tasks which would slow down the unit during recording.

One option is to record the data for playback and analysis at a slower rate.

Three sets of program selections for optimizing high–speed recording of data follow. All three versions assume that you are not loading in any program that will look at the TD or RD data, BCC, frames, packets, or EIA leads. All three versions record the data so that it can be run against more complex programs later on.

The fastest version records data without displaying the character data to the screen. The next version records data at maximum speed while still displaying data. The third version records data with personality packages loaded in the unit so that a protocol trace is displayed in real–time.

1. *Recording without displaying character data.* The first set of programming selections records all of a data line (including idle and time ticks) in bit–image data format. This is the fastest of the three recording scenarios.

   Beginning with the default screens, make the following selections:

   SETUP: LINE:

   Sync Char: delete both sync characters

   DISPLAY:

   Display Mode: NO DISPLAY

   RECORD:

   Capture Memory: user choice
   Data To Record: BIT IMAGE
   Stop at: user choice

   FEBUFF:

   Buffer Control Leads: YES or NO
   Time Ticks: OFF or ON with one second to one millisecond selected

   Using one–millisecond time ticks, the maximum normal recording speeds for each of the disk types on the Record setup screen are as follows:

Single Floppy Disk	64 Kbps
Multiple Floppy Disks	64 Kbps
Hard Disk	768 Kbps
RAM	768 Kbps
High–Speed RAM	2.048 Mbps (time–ticks off)

NOTE: Time–ticks are not supported above 768 Kbps.

TRIGGERS:

For SYNC format, the following triggers will keep the unit out of sync; the unit works at optimum speed when it is not in sync. Enter these conditions and actions on a Trigger Setup screen:

CONDITION: DTE 1 OF ☒ ("don't care")
ACTION: OUT_SYNC BOTH
CONDITION: DCE 1 OF ☒ ("don't care")
ACTION: OUT_SYNC BOTH

NOTE: When line rates approach the maximum recording speed, consider using high–speed RAM recording. See Section 7.5(A).

2. *Recording while displaying character data.* The second scenario records data at optimum speed while still preserving the character display.

Beginning with the default screens, make the following selections:

SETUP: LINE/RECORD:

Capture Memory: user choice for type of DISK
(do not select RAM recording)

FEBUFF:

Suppress the idle line pattern under **Idle Suppress** using the following patterns:

BOP: DTE : $^{7}_{E}$
DCE : $^{7}_{E}$
SYNC: DTE : $^{F}_{F}$
DCE : $^{F}_{F}$

On the Front End Buffer screen, turn off **Buffer Control Leads** and **Time Ticks**:

Buffer Control Leads: NO
Time Ticks: OFF
Frame Timestamps: OFF

LAYER:

Do not load any layer personality packages.

TRIGGERS AND PROTOCOL SPREADSHEET:

Do not load in any program that will look at the received data, BCC, or EIA leads.

3. *Recording while displaying character data or protocol trace.* The third setup version records data at maximum speed with layer personality packages loaded.

   Beginning with the default screens, check the following selections:

   SETUP: LINE/RECORD:

   Capture Memory: user choice for type of DISK
   (do not select RAM recording)

   FEBUFF:

   Suppress the idle line pattern under **Idle Suppress** using the following patterns:

   BOP: DTE : ^{7}E
   DCE : ^{7}E

   SYNC: DTE : ^{F}F
   DCE : ^{F}F

   On the Front End Buffer screen, turn off **Buffer Control Leads** and **Time Ticks**:

   Buffer Control Leads: NO
   Time Ticks: OFF
   Frame Timestamps: OFF

   LAYER:

   When running SS7 layer personality packages, load Layer 1 Compression package.

   TRIGGERS and PROTOCOL SPREADSHEET:

   Do not load in any program that will look at the received data, BCC, frames, packets, or EIA leads.

(B) Recording with SCSI Hard Disk Drives

Installing a SCSI (Small Computer System Interface) hard disk drive is one certain way to optimize high–speed record operation.

The maximum rate of recording bit–image data with time–ticks on, regardless of the type of hard drive is installed in your INTERVIEW 8000 unit, is 768 Kbps. However, with a SCSI drive in your INTERVIEW 8000 unit, selecting time–ticks *off* or either TD BITS or RD BITS selected in the **Tick Rate** field with time–ticks *on*, you can record a full–duplex 2.048 Mbps circuit at 100% line utilization; with either TD BITS or RD BITS selected, the other side of the line can use idle suppression. Additionally, application programs may be run simultaneously while recording 2.048–Mbps data.

Remote operation with an optional SCSI drive is permissible when recording at T1 speeds at up to 100% line utilization and at E1 speeds at up to 70% line utilization with idle suppressed.

2.11 Optimizing Analysis

The INTERVIEW 8000 Series can monitor data at high rates. If you want to monitor the data in real time, but still optimize speed, select **Time Ticks:** OFF and **Buffer Control Leads:** NO on the FEB Setup menu.

To achieve the maximum operation speed, the active spreadsheet program should *not* look at (TD or RD) data via string searches or (EIA) leads. This means that the following Layer 1, line–related C variables should not be referenced:

Event Variables:	Nonevent Variables:
evar_rcvd_char_rd	*rcvd_char_rd*
evar_rcvd_char_td	*rcvd_char_td*
evar_eia_changed	*current_eia_leads*
fevar_rcvd_char_rd	*previous_eia_leads*
fevar_rcvd_char_td	*l1_tick_count*
fevar_eia_changed	*rd_modifier*
	td_modifier

Other C variables at higher layers are either specific to various protocol packages or have meaning only if a frame has been passed up the layers.

Program output should *not* include performing routines such as *displayf*, *tracef*, or *printf* based on received events as these require a large amount of processing time (various code translations, colors, and string substitutions are often required).

In standard mode the INTERVIEW 8000 Series units can analyze 100% of the bandwidth up to 192 Kbps for the 8700 and 8800 units, up to 144 Kbps for 8600 units, and up to 64 Kbps for 8200 and 8100 units.

High Speed Frame Mode should be selected when the data rates are above the recommended rate of the unit you are using. In this mode the INTERVIEW analyzes 100% of the bandwidth to 2 Mbps by removing the overhead of the character data display (no character data display) and passing frames up to the higher layers.

> **NOTE:** Many of the application programs produced by General Signal Networks' AR Test Systems are are designed for extremely high frame rates (i.e., displaying traces for, analyzing statistics of, and emulating various protocols such as Frame Relay, SS#7, and SMDS).

3 Keyboard

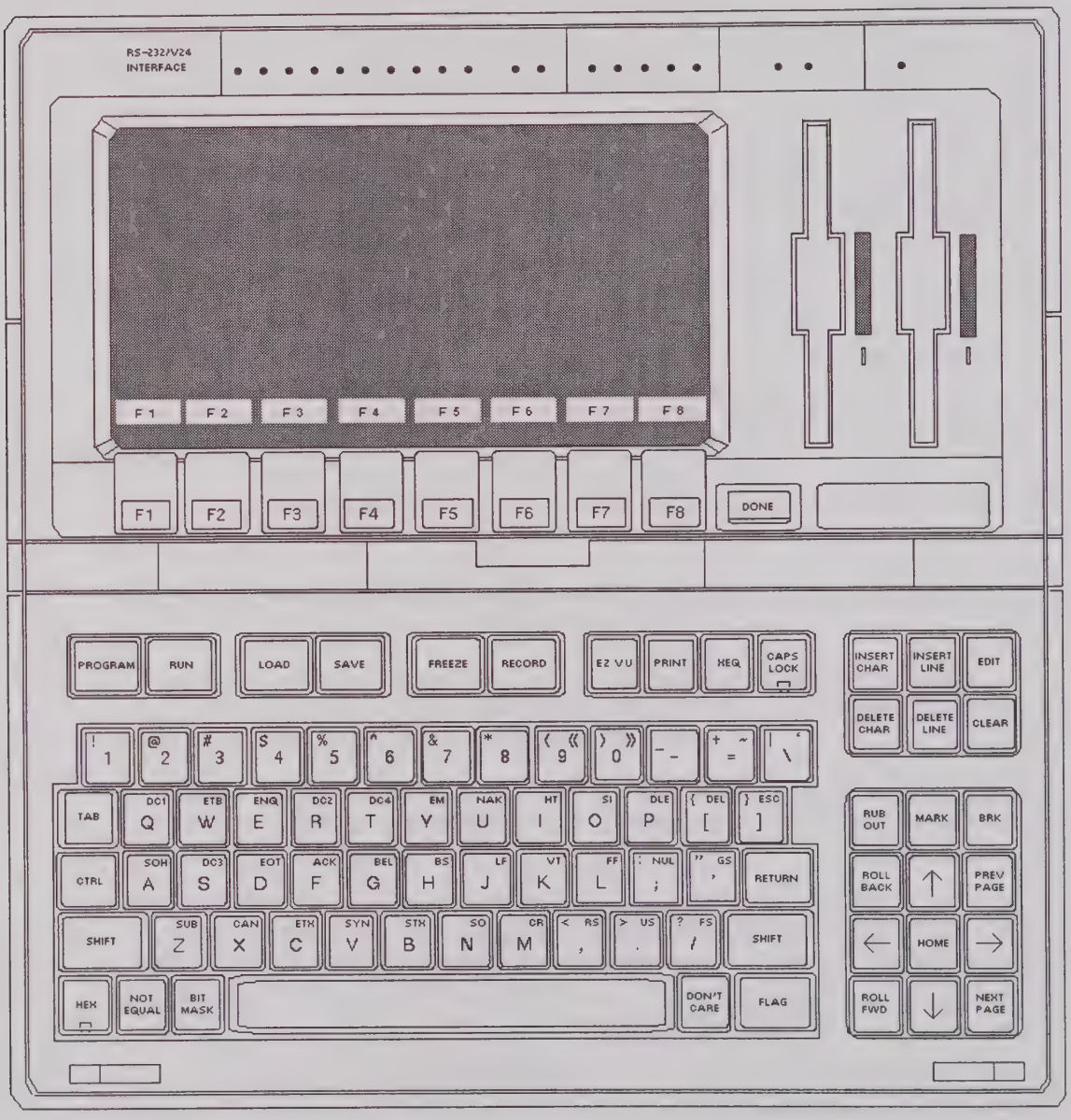


Figure 3-1 The INTERVIEW keyboard.

3 Keyboard

3.1 Hard Keys and Softkeys

The ninety−four keys of the INTERVIEW include nine keys (eight variable or *soft*keys and a DONE key) that are positioned not on the keyboard itself but rather above the keyboard on the front panel, just below the monitor. The softkeys are placed directly below their (changing) screen legends, in order that the users will not have to take their eyes off the screen to locate one of these keys.

In Program mode, softkeys can move the cursor around menu screens and change selections in menu fields. On the Protocol Spreadsheet screen, softkeys are an alternative to direct keyboard entry: programming levels from layer down to specific conditions and actions are softkey−selectable. In Run mode, the softkeys serve as an extended keyboard for functions such as changing the display (from data to protocol trace or to a statistics graphic, for example).

The keyboard itself consists of an ASCII keyboard and several special−function keypads, including a cursor keypad and an editing keypad that is used to modify all entries on the Protocol Spreadsheet screen as well as text entries on various menu screens. The ASCII keyboard includes five special, non−ASCII characters arranged on either side of the space bar that are used primarily to enter nonstandard characters in search strings.

Most keyboard functions are explained in detail elsewhere in the manual. The editing keypad, for example, is covered in Section 30, Editor. The present section is intended primarily to indicate the range of keys available when the operator has placed the unit in Program, Run, or Freeze mode.

> **NOTE:** For an explanation of the keys valid in the Easy View menus, refer to Section 4.

3.2 Programming Keys

Table 3-1 is a brief listing of some of the frequently used programming keys. The keyboards in Figure 3-2 and Figure 3-3 indicate all of the keys that are valid in Program mode.

Table 3-1
Frequently Used Programming Keys

EZ VU	If in Program mode, enter Easy View system. If in Easy View, exit to Program mode. Functional only in *TURBO* units when enabled on the Easy View Setup screen. See Sections 4, 19, and 20 for more information on Easy View operation.
PROGRAM	In Program mode, always returns to the master menu from any other menu or screen. (In Run mode, either returns to Program–mode master menu, returns to an Easy View menu, or loads, compiles, and runs program in *HRD/usr/user_intrf*. See Section 2.2(A).)
↑ ↓ ← →	Move cursor from field to field.
ROLL FWD ROLL BACK	Change rotating–field selections.
DONE	Completes an entry and tabs to next field.
HOME	Moves cursor to beginning of menu screen.
PREV PAGE	Goes back up to previous menu in program–menu tree.
XEQ	Loads and saves files; loads in protocol packages; executes other commands.
INSERT CHAR DELETE CHAR CLEAR RUB OUT	Edit text entries.

(A) ASCII Keys

The set of ASCII keys is used in Program mode to provide names for files and directories, to identify counters, timers, accumulators, and programming flags, to type in operator–defined messages called prompts and traces, to enter search and transmit strings on trigger menus, and to enter text on the spreadsheet screen. The ASCII character that appears on the screen is always the character shown on the keycap, with shifted characters shown at the top left corner of the keycap and control characters shown at the top right.

It is important to note, however, that there may be a difference between the display of keyboard input in Program mode versus Run mode. In Program mode, a particular keystroke (CTRL pressed in conjunction with =, for example) will consistently produce

the same character (" ~ ") in a data–entry field. In Run mode, a particular code–translate chart will be consulted and the character (" ~ " in our example) may not belong to the code set and therefore may be missing on the chart. Such a character cannot be trapped, transmitted, or displayed in Run mode: it is "untranslatable" (see Note, Appendix D1). Run–mode encoding of keyboard input is given in Appendix D1, Keyboard–to–Hex Translation.

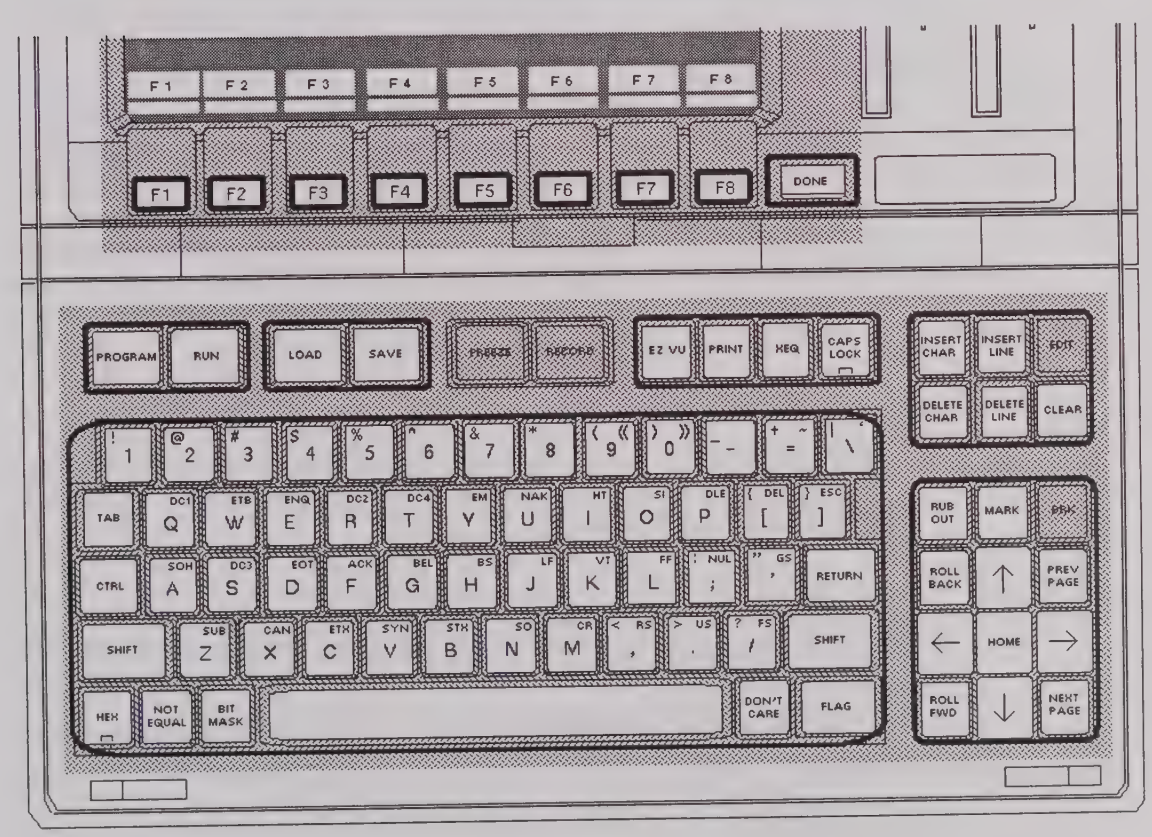


Figure 3-2 Menu–screen keys.

Two characters on the ASCII keyboard are part of the spreadsheet programming language but do not belong to any standard code set. They are « and ». These characters are produced by the combination of CTRL and 9 or 0 and can be entered only on the Protocol Spreadsheet, and in the **Xmit:** field on Trigger Setup menus. They are never transmitted.

For examples of how these double–parens are used, see Section 29, Constants, and Section 33, Strings.

Three standard ASCII characters have special meaning on the Protocol Spreadsheet. The double–quotation mark (") opens or closes a text list or string: see Section 33, Strings. The backslash (\) is an escape character. And the ASCII space produced by the space bar indicates the completion of a "keyword" (or "token") and normally changes the rack of programming softkeys along the bottom of the screen.

[RETURN] does not produce an ASCII control character in the INTERVIEW. It is always a cursor–movement key. Press [CTRL]–[M] to enter the ␍ character.

(B) Special Characters

[NOT EQUAL], [BIT MASK], [DON'T CARE], [FLAG], and [SHIFT]–[FLAG] produce entries only in search fields in trigger conditions and in spreadsheet conditions. [HEX] is an on–off function. In the "on" condition it lights the LED on the keycap and produces hexadecimal entries both in search fields and in transmit strings. The function works as follows: with the hex function off, the key sequence [A], [A], [HEX], [7], [F], [7], [F] yields the screen entry of AA$^{7}_{F}$ $^{7}_{F}$. Hex is also a valid data–entry function in the **Sync Char**, **Outsync Char**, and **Xmit Idle Char** fields on the Line Setup menu, in text–entry fields on the BCC Setup menu, and in the **Preamble** and **Sync Pattern** fields on the BERT Setup screen.

(C) Load/Save

[LOAD] and [SAVE] call up the File Maintenance menu directly from any other programming screen. [LOAD] selects **Command:** LOAD on the File Maintenance menu, while [SAVE] selects **Command:** SAVE.

(D) EZ VU

[EZ VU] is a toggle key. Use it to move back and forth from Program–mode screens to Easy View.

The user can enable/disable the [EZ VU] key on the Easy View Setup screen. (See Section 19.)

(E) Shift–Save

Press [SHIFT]–[SAVE] to save the configured Printer Setup or Easy View Setup menu. The Printer Setup is saved to a file called *print_setup* in the */sys* directory of the boot–up disk. The Easy View setup is saved to a file called *ezview_setup* in the */sys* directory of the boot–up disk.

(F) Print

Press [PRINT] to send the program menu (or spreadsheet page) that is currently displayed to a serial printer attached to the INTERVIEW. [CTRL]–[PRINT] will print the full set of programming menus (or the full spreadsheet program).

(G) Execute

[XEQ] is the key that performs the actual load or save. [XEQ] also performs other commands on the File Maintenance menu as well as various functions on the Disk Maintenance, Date/Time Setup, and Layer Setup screens. On the Protocol Spreadsheet, [XEQ] may be used to perform a "write" operation from the screen to a file, or a "read" operation from a file to the screen. See Section 30, Editor.

(H) Mark

[MARK] has a special function on the File Maintenance Screen. When you have cursored down to a particular file in the File Maintenance directory, [MARK] selects that file for the next operation. Several files may be "marked" prior to the execution of a command. The [MARK] key also unmarks rows in the directory that have been marked already.

[MARK] has a "program tab" function on the Protocol Spreadsheet: see (L), below.

(I) Editing Keys, Menu Screens

The editing keys that operate within text–entry fields on menu screens are shown in Figure 3-2. [INSERT CHAR] inserts a space at the cursor position. A character can then be typed over the space. [DELETE CHAR] deletes the character that is under the cursor. Characters in the same field to the right of the cursor move left to fill the column vacated by the deleted character. [CLEAR] removes the entire data entry.

[RUB OUT] deletes the character just to the left of the cursor position. The cursor moves left as successive characters are deleted. Use [RUB OUT] to correct a typographical error on the most recent keystroke (or the last several keystrokes).

[CTRL]–[CLEAR] restores the default selections for the entire menu screen.

On the multipage tabular and graphic statistics screens, [INSERT LINE] adds a blank statistics row above the cursor. [DELETE LINE] removes the row that has the cursor.

(J) Editing Keys, Spreadsheet Program

These keys are shown in Figure 3-3. The operation of [DELETE CHAR] and [RUB OUT] is the same as on menu screens, except that here the use of these keys is not restricted to particular fields.

[INSERT LINE] adds a blank line above the cursor. [DELETE LINE] removes the line that has the cursor. [CLEAR] erases the cursor character and everything to the right of the cursor on a given line. [EDIT] enables/disables the spreadsheet editor. Refer to Section 30, Editor, for editing softkeys and their functions.

There is a separate insert *mode* in the spreadsheet editor. This mode is invoked by two keys, INSERT CHAR and INSERT LINE. When the mode is enabled, the word <Insert> appears at the top left of the spreadsheet screen. In insert mode, the operator types in a block of data while succeeding text is pushed forward with every keystroke.

Press INSERT CHAR a second time to exit *insert* mode and return to *overstrike* mode.

NOTE: INSERT LINE is not an alternate–action key that toggles the spreadsheet from *insert* to *overstrike* mode. Only INSERT CHAR accomplishes this function.

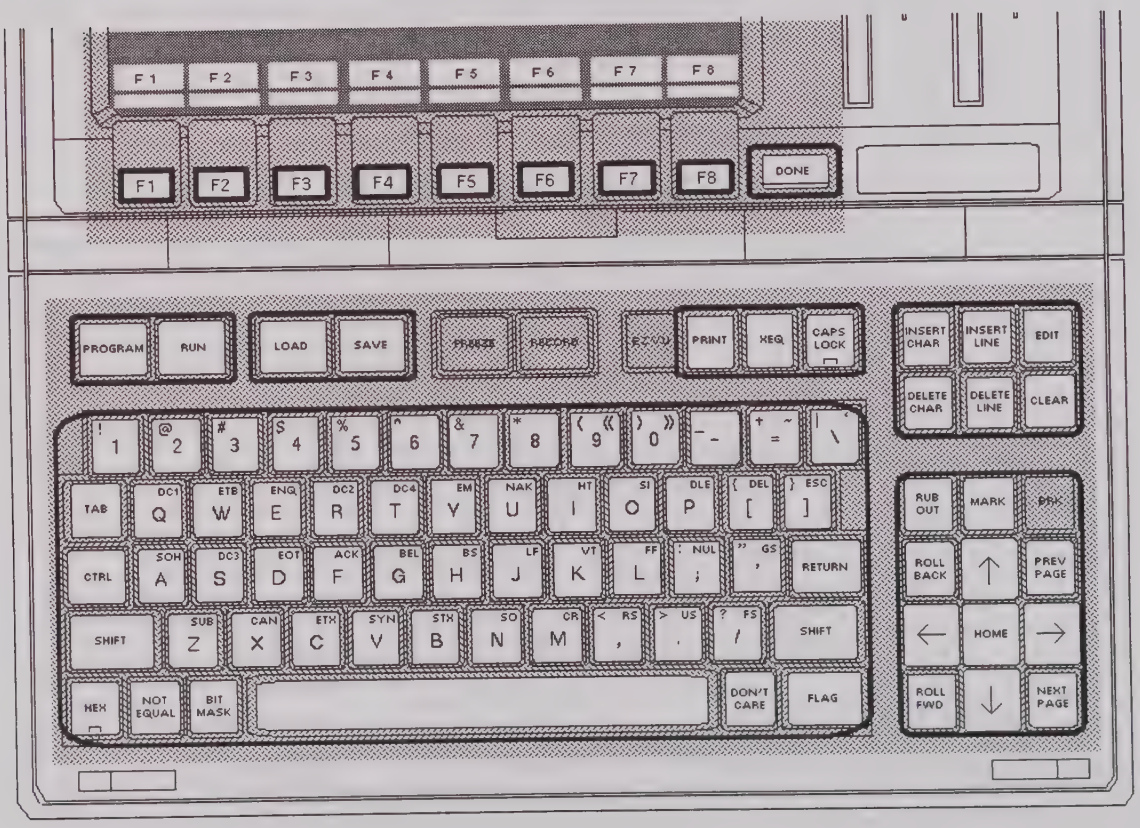


Figure 3-3 Spreadsheet keys.

(K) Cursor Keys, Menu Screens

See the bottom right of Figure 3-2. ↑ and ↓ move the cursor from line to line on programming menus (including triggers). SHIFT–↑ moves the cursor to the first field in the menu. SHIFT–↓ places the cursor in the last field in the menu. → and ← move the

cursor from field to field on the same line. In text–entry fields, they move nondestructively right and left over text that has already been entered. [ROLL FWD] and [ROLL BACK] rotate the selections at the cursor position.

[RETURN] and [DONE] are field oriented. They move the cursor down or to the right, depending on where the next menu field is located.

Whenever a programming menu is first entered, the cursor is in the "home" position. [HOME] moves the cursor back up to this position. Home may be the first field on the menu, or it may be a softkey rack that selects among menus or among fields on a lengthy menu. Several screens have more than one home position. Additional positions are accessed consecutively when you "home upwards" from below. See, for example, the Line Setup or Trigger or Misc Utilities screens.

When you are accessing a programming screen (except the main–menu screen or the two Statistics menus), the [PREV PAGE] key will return you to your previous menu. You may use this key to backtrack through several previous screens.

On the two multipage Statistics screens, [PREV PAGE] and [NEXT PAGE] cause a page–by–page scroll.

(L) Cursor Keys, Spreadsheet Program

In a spreadsheet program, the cursor–arrow keys move the cursor by one column or one row. [RETURN] moves the cursor to the current indent position on the next line. [ROLL BACK] and [ROLL FWD] cause the screen to scroll one line at a time, while [PREV PAGE] and [NEXT PAGE] cause a page–by–page scroll. Roll and page keys do not affect the cursor.

"Home" is the top left of the current screen. To cursor to the beginning of the spreadsheet program, press [SHIFT]–[↑]. Press [SHIFT]–[↓] to go one line below the last line of the program.

To cursor to the beginning of a given line, press [CTRL]–[←]. To "express" the cursor to the end of the data–entry on a line, press [CTRL]–[→].

To cause the cursor to skip forward from keyword (or "token") to keyword, press [SHIFT]–[→]. [SHIFT]–[←] causes a reverse skip.

Any cursor movement across keyword boundaries will change the softkey–option rack along the bottom of the screen.

[MARK] may be used as a kind of "program tab" to mark a place in the Protocol Spreadsheet where the cursor will automatically revert on command. With the cursor in a position you will want to return to quickly and conveniently, press [MARK] followed by a number key—[2], for example. Then go ahead and move the cursor any distance in either direction from the marker.

When you wish to return to the "mark 2" position, press [SHIFT]–[MARK] followed by [2]. Up to ten numbered cursor–markers may be reserved in the spreadsheet program.

(M) Softkeys and DONE

Depending on which menu field has the cursor, all selections are mapped to softkeys at various times. On menu fields, the DONE key has the same function as RETURN: it moves the cursor to the next field. So there is a way to move around the menu and make selections without using the cursor–arrow and other keyboard keys.

When the operators are using softkeys to enter a spreadsheet program, they use DONE to mean, "Escape to the previous (higher) level of programming." Figure 3-4 illustrates how two depressions of the DONE key can take the program from a specific EIA condition all the way to the highest level of softkeys.

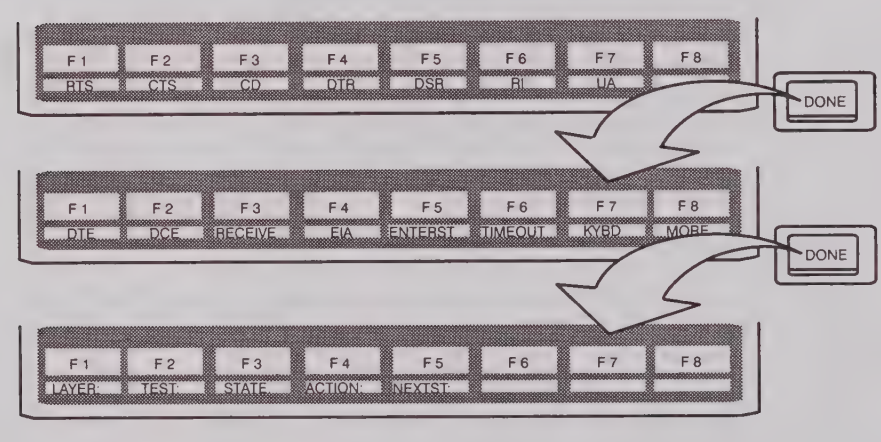


Figure 3-4 Moving to a higher rack of softkeys on the Protocol Spreadsheet.

3.3 Real–Time Keys

The keyboard in Figure 3-5 indicates the keys that are valid in Run mode when the data is displayed in real time (rather than frozen). In addition to the keys highlighted in this figure, the C programmer may use the variable *keyboard_any_key* to monitor input from every key except EDIT, CAPS LOCK, and DONE. See Section 68, Other Library Tools.

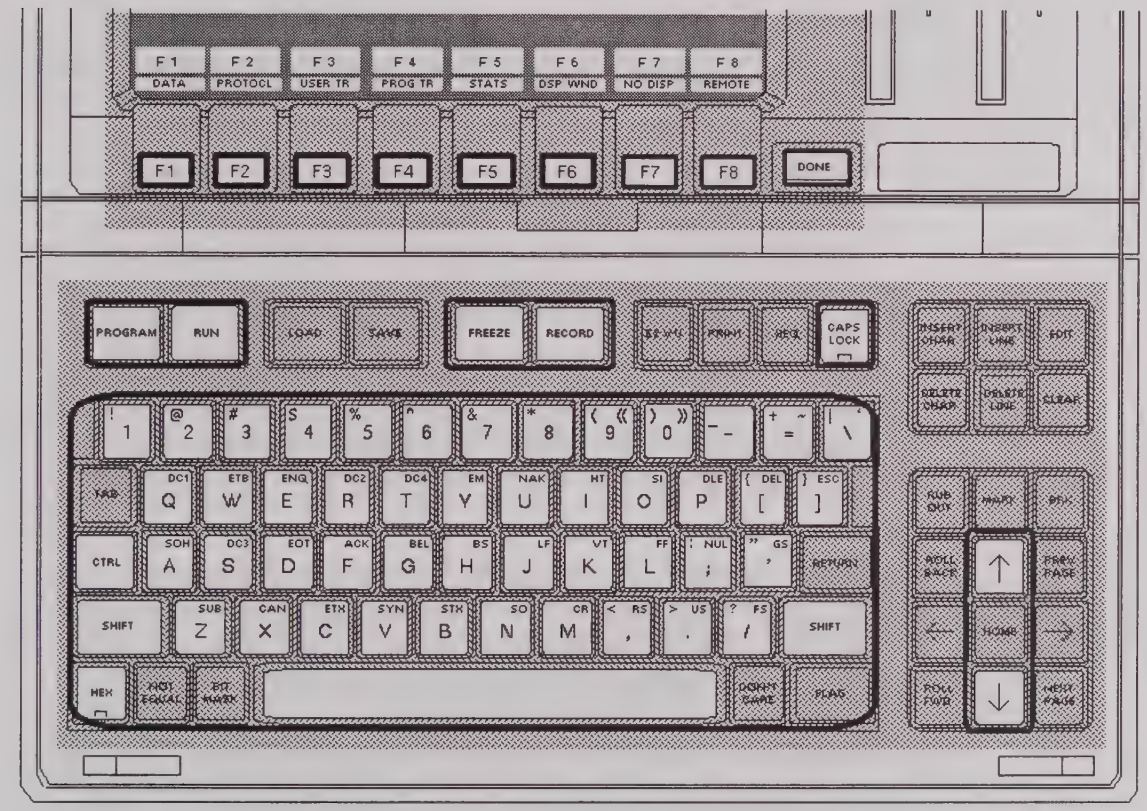


Figure 3-5 Real−time keys.

(A) Hex

In normal Run−mode operation, the [HEX] key controls an on/off decoding function that converts all the data on the screen to hexadecimal. Note that the screen−decoding function of [HEX] does not light the LED on the keycap. Only the data−entry function turns on the LED.

(B) Freeze

[FREEZE] controls an on/off function that freezes the screen display. For line data, program activity and bit−image recording continues. Character recording will not continue since the display buffer (character RAM) is frozen.

A similar freeze function can be activated when a **Capture:** BOTH OFF action is performed by a trigger. The difference is that while the manual (keyboard) freeze permits you to scroll through the data buffer, trigger freeze does not.

(C) Record

For line data, RECORD controls a manual start/stop function that records data according to the parameters selected on the Record Setup menu. For disk data, press RECORD to suspend/resume playback.

> **NOTE:** Although playback is immediately suspended when you press RECORD, the screen display continues until the character buffer's contents are fully displayed. (For bit–image data, the FIFO must empty.) At slower playback speeds, you may notice a slight delay before the display actually freezes.

Notice the record/playback status field next to the block number field in the status area of the Run–mode display screens. See Section 6.2. The initial status indicator displayed in this field is determined by Line and Record Setup selections. See Sections 5.2 and 7. It is subsequently controlled via the RECORD key.

(D) Cursor Keys

When a multipage Statistics display (tabular or graphic) is presented in Run mode, ROLL BACK and ROLL FWD cause the rows of values (or bars) to scroll up or down one line at a time, while PREV PAGE and NEXT PAGE cause a page–by–page scroll.

↓ and ↑ serve two purposes. When the Run–mode display screen is the Display Window, ↓ and ↑ are under the programmer's control. (Refer to Section 68, Other Library Tools, for information on keyboard variables and the *send_key* routine.) For example, these keys could be used to move from field to field on a menu created in the Display Window. For all other Run–mode screens, ↓ and ↑ control the playback speed of data from a disk. ↓ slows the data speed by half, while ↑ doubles the current speed.

> **NOTE:** When you use the ↓ and ↑ keys to control the data speed during playback, an alarm will sound if you attempt to slow the data below the minimum speed required or speed the data beyond the maximum speed allowed.

(E) Softkeys

In Run mode, the softkeys will change the display selection. Selectable display modes include character data, character data plus control–lead timing, protocol trace, program trace of state–to–state movement and of user–entered messages called "traces," an application–specific "display window," statistical tabulation or graphic display of counters and timers, and remote operation from a PC or another INTERVIEW.

There is also a Run–mode softkey selection called NO DISP. This suppresses the writing of data to the screen (though not to the screen buffer). See Section 6.11 for an explanation of this display mode.

3.4 Freeze–Mode Keys

The keyboard in Figure 3-6 indicates the keys that are valid in Run mode when the data display has been frozen.

In addition to the keys highlighted in Figure 3-6, the C programmer may use the variable *keyboard_any_key* to monitor input from every key except [EDIT], [CAPS LOCK], and [DONE]. See Section 68, Other Library Tools.

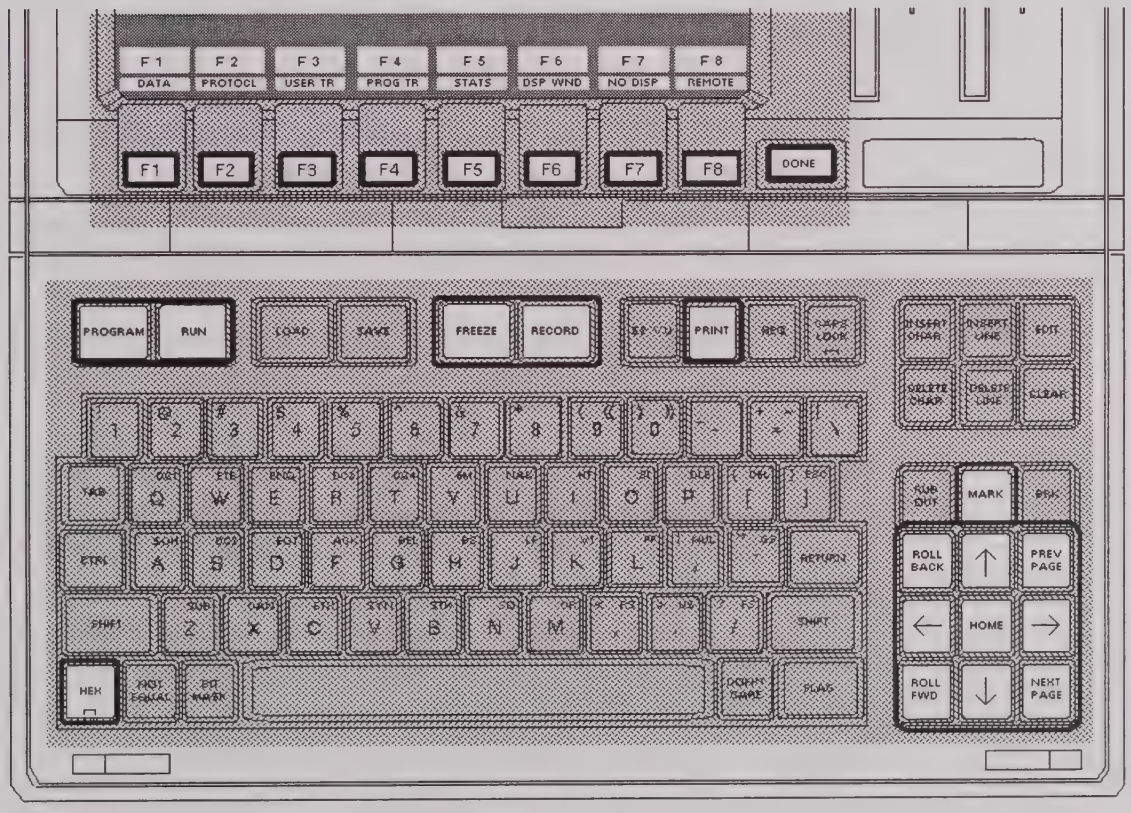


Figure 3-6 Freeze–mode keys.

(A) Hex

In Freeze mode, [HEX] controls an on/off decoding function that converts all the data on the screen to hexadecimal.

(B) Freeze

[FREEZE] will also unfreeze a frozen display. For disk data, it will resume playback and program activity.

(C) Record

The recording of live data can be stopped and started even while the display is frozen. The [RECORD] key will not resume playback of disk data that was frozen via the [FREEZE] key.

(D) Print

Press [PRINT] to send the current frozen screen to a serial printer attached to the INTERVIEW. [PRINT] together with [CTRL] will send the entire data buffer.

(E) Cursor Keys

The cursor keys work on the frozen data buffer the same way they work on the spreadsheet screen. Cursor–arrow keys move the cursor by one column or one row. [ROLL BACK] and [ROLL FWD] cause the screen to scroll one line at a time, while [PREV PAGE] and [NEXT PAGE] cause a page–by–page scroll. Roll and page keys do not affect the cursor.

[HOME] will move the cursor to the first (and oldest) character in the character (screen) buffer. When [HOME] is pressed together with [CTRL], the screen and cursor move to the last (or newest) character in the buffer.

When a multipage Statistics display (tabular or graphic) is presented in Freeze mode, [ROLL BACK] and [ROLL FWD] cause the rows of values (or bars) to scroll up or down one line at a time, while [PREV PAGE] and [NEXT PAGE] cause a page–by–page scroll.

(F) Mark

The [MARK] key enables/disables the cursor timing feature of three Freeze–mode data displays (all except dual–line). When you first enter Freeze mode, cursor timing is disabled. Press [MARK] to enable it.

The [MARK] key is also used to mark the position of an event in the display buffer. The marked character is replaced by the [M] symbol. Each time you press [MARK], you move the marked position—i.e., you redefine the event.

To disable the cursor timing display, return the cursor to the marked character and press [MARK].

For additional information on cursor timing, see Section 6.4(E).

(G) Softkeys

All softkeys are valid in Freeze mode and serve the same functions as in real time.

3.5 Remote Control Keys

The INTERVIEW keyboard–to–PC conversion for remote–control operation is shown in Table D4-1.

4 Easy View

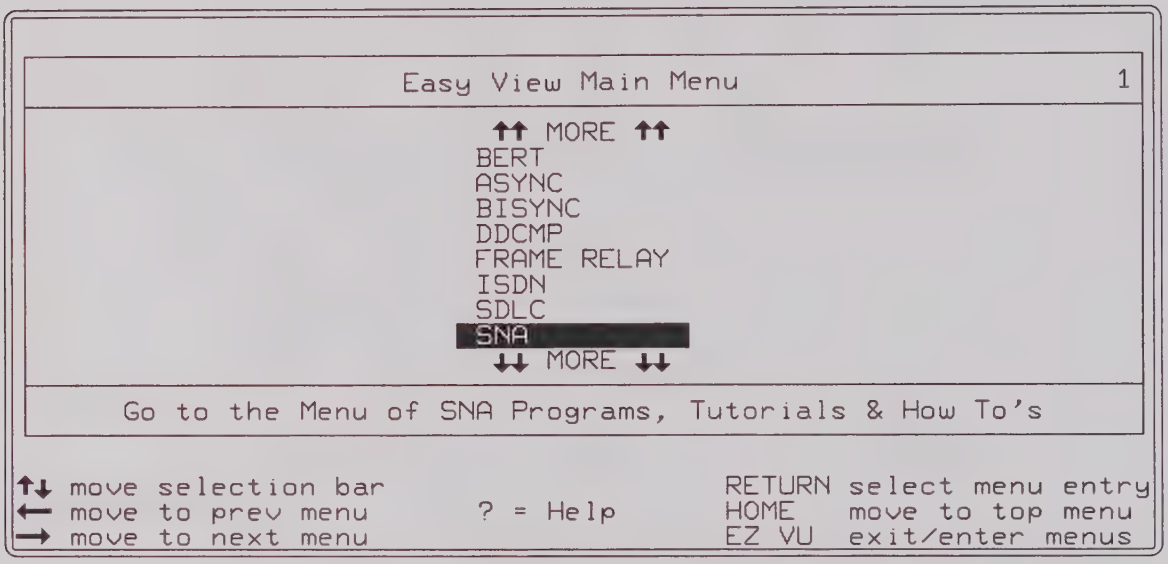


Figure 4-1 The main menu in the Easy View system.

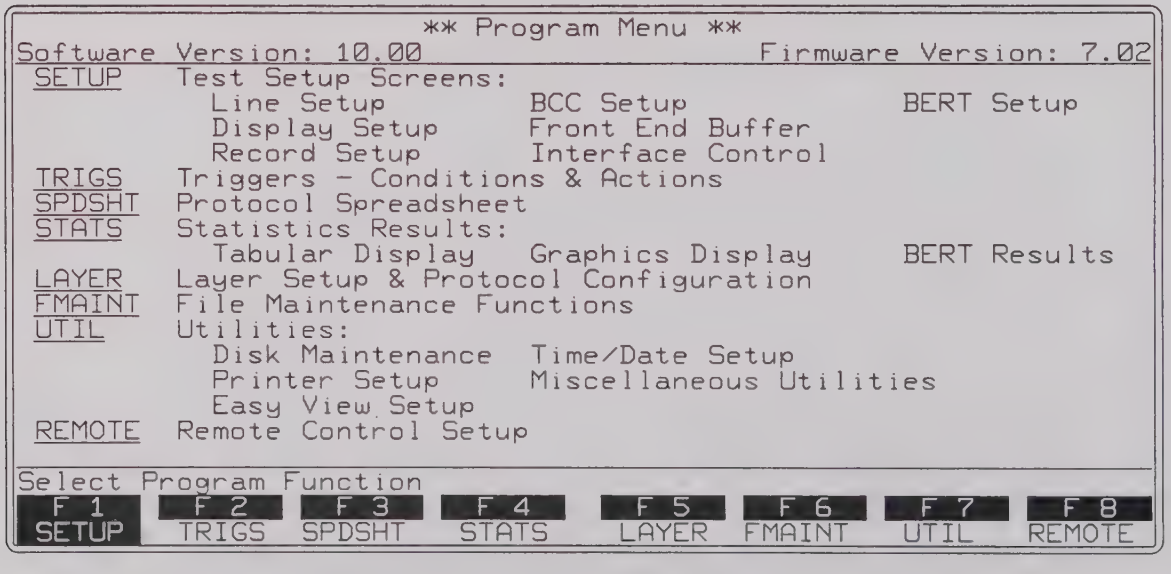


Figure 4-2 The main Program menu screen in Program mode.

4 Easy View

This section provides an overview of the Easy View system. It describes the appearance of screens and the keys used to interact with those screens. Use the menus in Easy View to load and execute programs quickly or to access help information or tutorials about INTERVIEW screens, menu selections, and protocols.

The Easy View Operating System requires software revision 10.00, or higher.

4.1 Booting Up

By default, *TURBO* units automatically enter Easy View after boot−up. There is a five−second pause at the power−up screen before you see the first Easy View menu, shown in Figure 4-1. Use this pause to obtain any pertinent information you need from the power−up screen. If you want to bypass this brief timeout and enter Easy View directly, press [PROGRAM]. From this point on, toggle between Easy View and Program mode by pressing [EZ VU].

> **NOTE:** If the boot−up disk contains the file */usr/default* or the hard disk contains the file */usr/user_intrf*, the INTERVIEW does not automatically enter Easy View following boot−up, even if it is configured to do so. These are user−created programs that automatically start the INTERVIEW in Run mode. See Sections 2.1 and 2.2 for more information on these files.

The INTERVIEW may be configured to enter Program mode instead of Easy View following boot−up. Go to the Easy View Setup screen and modify (and save) the configuration. See Section 19.

You can reset the INTERVIEW in Easy View or in Program mode directly from the keyboard. Press the [CTRL]−[SHIFT]−[DELETE CHAR] or [CTRL]−[SHIFT]−[DELETE LINE] key combination to force the INTERVIEW to perform the self−test and booting sequences without turning the unit off. This reset is similar to that of a PC in that it is not a hardware reset. For hardware resets, you must power the unit off and then back on.

4.2 Entering and Exiting Easy View

If you entered Easy View automatically after boot–up, you may exit (from any of its screens) by pressing EZ VU. When you exit Easy View after boot–up or after you run a program from an Easy View menu, the main Program menu screen shown in Figure 4-2 is displayed. In general, however, whenever you exit Easy View, you are returned to the same location in the Program–mode screens where you were when you accessed Easy View.

As long as the EZ VU key is enabled, reenter Easy View by pressing EZ VU again. If Easy View menu information is being kept in memory (see Section 19), you are returned to the same location in the Easy View menus where you were when you exited. Repeatedly pressing EZ VU moves you back and forth between Easy View and a Program–mode screen. Use this feature to access help information about the selections you see on Program–mode menus. Section 4.4(D) explains how to find help information.

Press CTRL–EZ VU to force the INTERVIEW to enter Easy View and reload its menus from the hard disk. This feature may be particularly useful if you have copied a new version of the master menu file from a floppy disk to the hard disk and you want to load the new version.

> **NOTE:** If you press EZ VU while the INTERVIEW is performing an extended disk or file operation from the Disk Maintenance or File Maintenance screen—such as copying files, formatting a disk, or obtaining the contents of a directory—an alarm sounds and the screen remains unchanged. Once the operation has been completed, the EZ VU key functions normally.

4.3 Easy View Menus

The Easy View menus are organized into a tree–structured hierarchy. See Figure 4-3. The first time you enter Easy View following boot–up, the menu at the root of the tree, the Easy View Main menu, is displayed. See Figure 4-1. The root is level one of the hierarchy. Any menus entered directly from the root menu are level two. Menus entered from level two menus are level three, and so on.

The Easy View Main menu is organized primarily by protocol, as shown under "Level 1" in Figure 4-3. Other menu items provide: programs for autoconfiguration (AUTOMON) and BERT testing (BERT); access to specific information about various operations, screens, and menu selections of the INTERVIEW (OPERATION); a list of Easy View utility programs (UTILITIES) such as file transfer; and updates on available options and new product information (AR NEWS).

When you select a protocol from the Easy View Main menu and press RETURN, a menu of Program Types, Tutorial(s) & How To's is presented. Refer to "Level 2" in Figure 4-3. From this menu, identify which kind of program you want to run for your selected protocol, typically monitor, statistics, emulation, or conformance (for

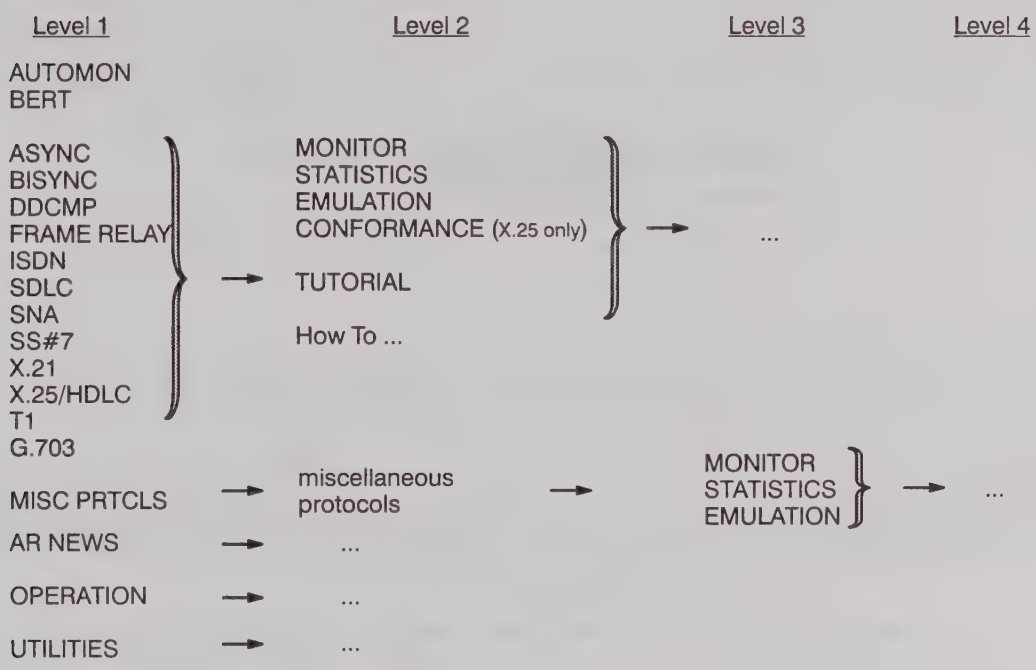


Figure 4-3 The Easy View system has a tree–structured menu hierarchy, similar to the one shown above.

X.25 only). In addition, you can access tutorial or how–to information about the protocol. Tutorials provide general information about the selected protocol. How To's guide you in the setup and operation of the INTERVIEW according to the protocol environment.

If you select MONITOR, STATISTICS, EMULATION, or CONFORMANCE (if present) from a Program Types, Tutorial(s) & How To's menu and press RETURN, a level–three menu lists the available programs.

(A) Menu Format

Easy View menu screens have the format shown in Figure 4-4.

1. *Master title.* The top line of the screen may contain a master title. This title is centered at the top of all Easy View menu screens. In Figure 4-4, it is blank. See Section 20.5 for information on defining a master title.

2. *Menu title.* Each menu has an individual title centered in the top box of the screen.

3. *Menu level.* A number indicating the menu's level in the menu hierarchy is displayed at the right end of the menu title line.

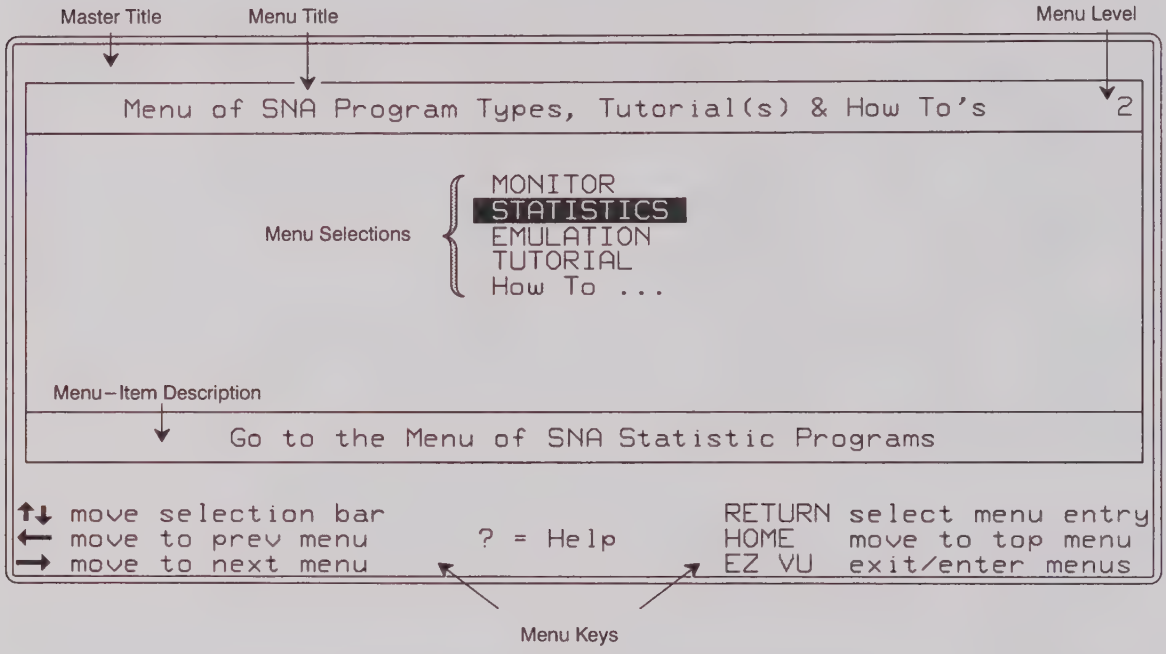


Figure 4-4 A second–level Easy View menu screen. To access this menu, select *SNA* on the Easy View Main menu and press RETURN.

4. *Menu selections.* The center box on the screen displays the menu items. A reverse–video cursor bar highlights the selected menu item. In Figure 4-4, STATISTICS is selected for SNA protocol.

   Not all menus contains selections. These empty menus have been included for the addition of optional AR programs. You may also use them for your own programs.

   Although a menu has no maximum number of selections, each screen displays no more than eight menu items at a time. If there are more than eight entries in the menu, the word "MORE" is displayed in a position that indicates where the additional items fall in the list. Refer again to Figure 4-1.

5. *Menu item description.* Each menu item can have an associated line of descriptive text, centered in the bottom box on the screen. This text is displayed automatically when the selection bar is over the menu item. The text shown in Figure 4-5 describes the program that will be run if you select Troubleshoot Host=DTE and press RETURN.

   If the selection bar is over a program item that is available as an option, but not currently installed in the INTERVIEW, a message to that effect is displayed.

6. *Menu keys.* At the bottom of each menu screen is a list of the keys you may use to interact with the menu.

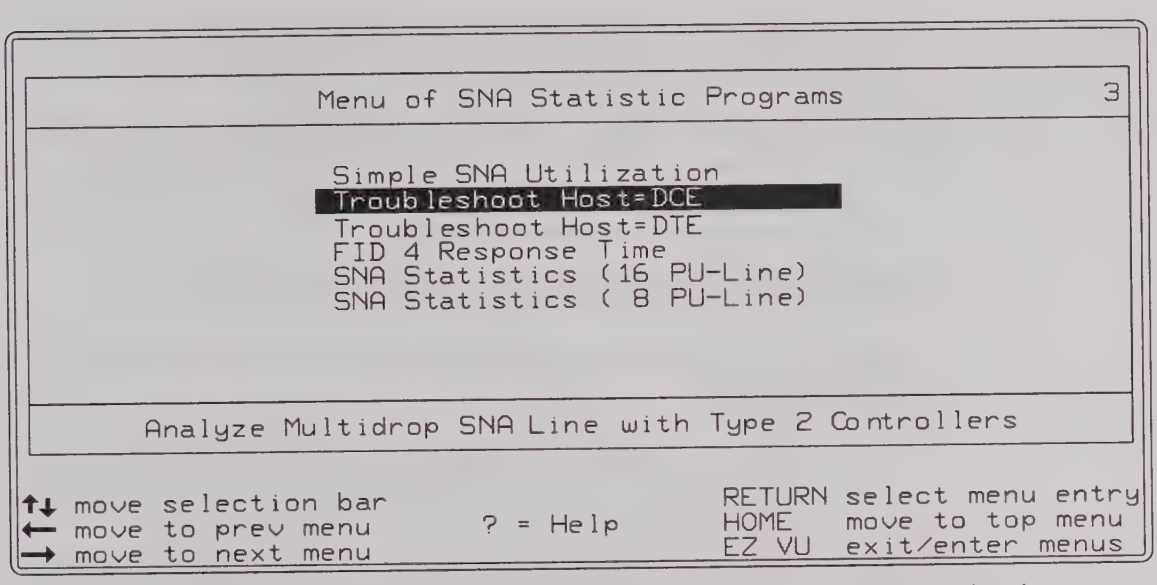


Figure 4-5 Statistic program selections for SNA protocol. To access this menu from the screen shown in Figure 4-4, position the selection bar over *STATISTICS* and press RETURN.

(B) Menu Keys

Several keys are used to interact with the Easy View menus.

The most commonly used keys are displayed at the bottom of each menu screen along with brief descriptions of their functions.

↑ Moves the selection bar up one menu item on the screen. If the selection bar is on the top menu item and there are prior items out of view, the list scrolls down one item and the selection bar remains in position.

↓ Moves the selection bar down one menu item on the screen. If the selection bar is on the bottom menu item and there are additional items out of view, the list scrolls up one item and the selection bar remains in position.

← Returns to the previous, or "parent," menu in the menu hierarchy.

→ If the action associated with the menu item under the selection bar is go to a menu, you may use the → key to move down one menu in the menu hierarchy. Otherwise, the → key has no effect.

? Accesses help information available about a selected menu item. The help file presents additional information about the selected menu item.

RETURN Executes the action associated with the menu item under the selection bar: move to another menu, load and run a program, or view a text file. The

RETURN key can be used instead of the → key to move to another menu. RETURN is the only means available to run a program or view a text file.

HOME — As you move through the menu hierarchy, Easy View records the selections you make. Use the HOME key to move immediately to the top of the root menu and clear your selections from memory. The system now acts as if it is entering each menu for the first time.

EZ VU — If enabled, toggles between Easy View and Program mode.

There are additional keys that operate on Easy View menus which are not displayed on the menu screens:

PREV PAGE — Use this key to highlight the first menu item on the screen. If there are additional entries on a preceding page of the menu and you press PREV PAGE again, the cursor is positioned over the first entry on the previous page.

NEXT PAGE — Use this key to highlight the last menu item on the screen. If the menu has more than eight items and you press NEXT PAGE again, the cursor is positioned over the last entry on the next page of the menu.

4.4 Using Easy View

The first time you enter a menu, the initial eight menu items are displayed on the screen. A reverse–video selection bar is positioned over the top item. Use the cursor keys to move the selection bar over a menu item and press RETURN to perform the action associated with that entry.

When you select a menu item, one of three actions occurs: another menu is displayed, a program is loaded and run, or a text file is presented for viewing on the screen.

(A) Moving through the Menus

As you move through the menu hierarchy, Easy View records all the selection paths you make from its menus. When you return to each menu, its previous state is restored. This feature allows you to retrace any selection path up (or down) the menus by repeatedly pressing ← (or →).

Press HOME to clear the record of menu selections and return to the Easy View Main menu. The system now acts as if it is entering each menu for the first time.

(B) Running Programs

Once you have made a program selection on an Easy View menu, press RETURN to load, compile, and run the program.

By default, the message shown in Figure 4-6 is displayed when you select a menu item whose action is to run a program. If you do not want to have this warning message presented for the rest of your working session (i.e., until the INTERVIEW is turned off), disable it by pressing D. Then decide whether to run the program or return to the menu without running it.

NOTE: To disable the warning message completely, go to the Program–mode Easy View Setup screen. See Section 19.

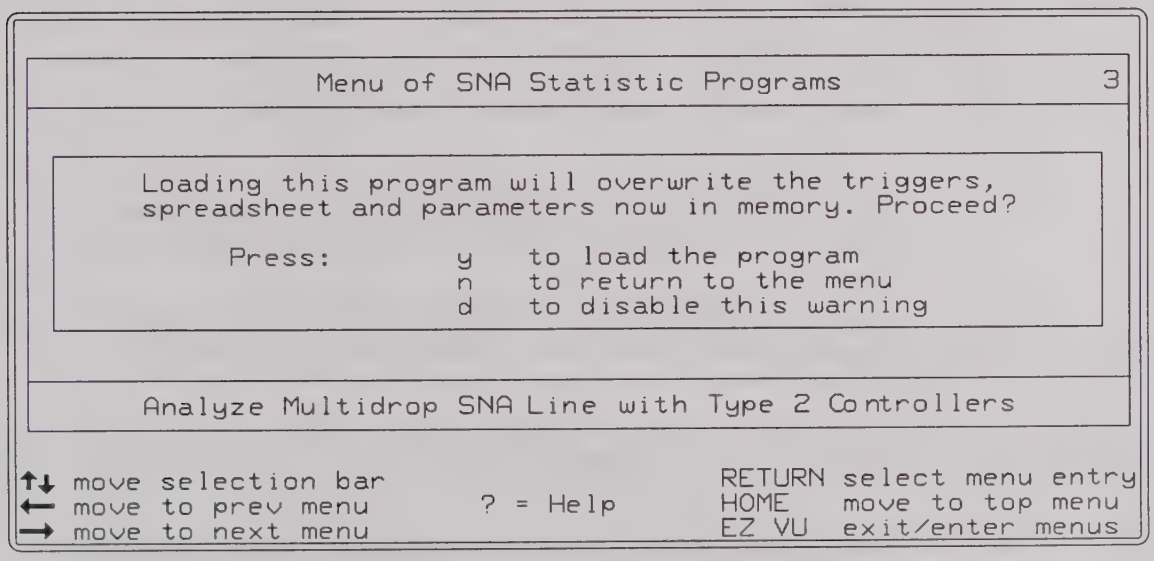


Figure 4-6 A warning message is presented before running a program. This program was run by selecting *Troubleshoot Host=DTE* from the menu shown in Figure 4-5 and pressing RETURN.

If you decide to proceed, press Y. The screen shown in Figure 4-7 appears. There is a brief pause at this screen while the program is loaded. Some of the line setup parameters for Easy View programs can be changed on–line before the program is actually run. You can elect to modify these parameters by pressing EDIT from this screen. See Section 4.4(C). For now, we will leave the parameters intact, so do *not* press EDIT.

The INTERVIEW attempts to compile and run the program. If an error occurs at any step in this process, an error message is displayed and you are automatically returned to the menu screen from which the selection was made. (You are *not* automatically taken to the spreadsheet screen if a compilation error occurs—as you are when you run a program from Program mode.)

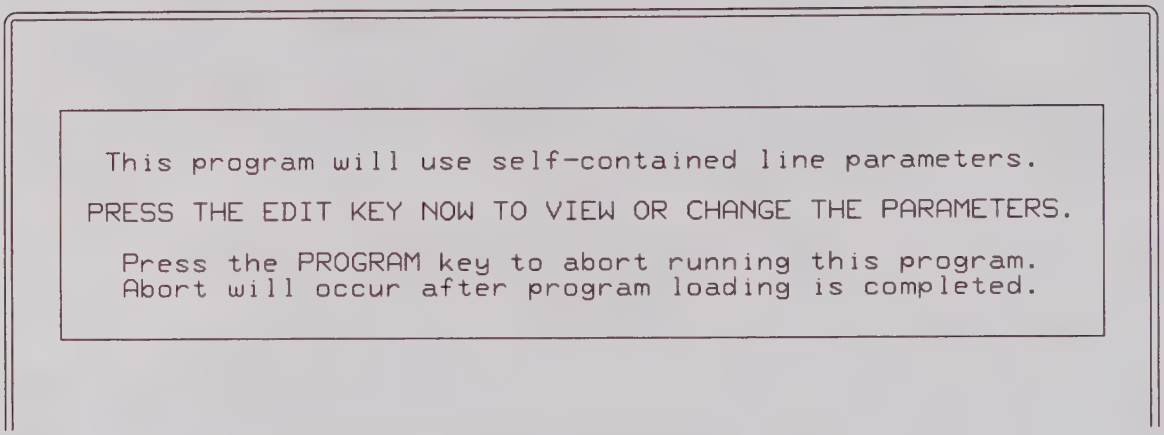


Figure 4-7 Decide whether you want to view and/or change program line parameters.

If you exit the program by pressing PROGRAM (and the Easy View Setup screen described in Section 19 shows **Keep Easy View Menu Information in Memory:** YES), you are automatically returned to the menu screen from which the program was launched.

Figure 4-8 shows the first Run−mode display for an SNA statistic program that was selected and run from Easy View. (The program was selected in Figure 4-5).

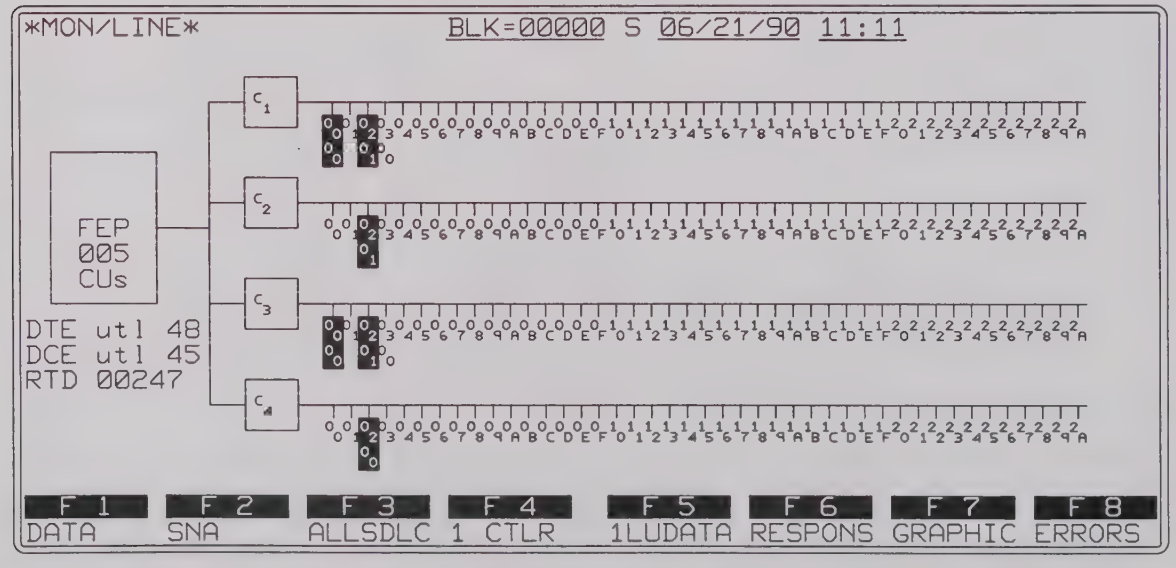


Figure 4-8 Run−mode display of controller and LU activity on an SNA multidrop line.

(C) Changing Line Parameters

Some of the line setup parameters can be changed on−line before the program is actually run. These parameters include **Data source**, **Disk**, **Disk Block No.**, **File**, **Clock**, **Speed**, and **NRZI**. (To change other setup parameters, see Section 4.5(A).)

The default parameters associated with an Easy View program (and included in the same linkable−program or source file) are its *self−contained* line parameters. If changes are made to any of the parameters, the edited set is referenced as *Easy View* line parameters.

The first time you elect to run a program in any INTERVIEW session, the program is set to use the self−contained parameters. Refer again to Figure 4-7. If you want the program to run with these parameters, do nothing. Once the program loads, it will compile and run. If you want to view the settings or make changes, press [EDIT] before the program is finished loading. The self−contained parameter settings are displayed.

Self−contained line parameters for Easy View programs typically, although not always, have the following default settings:

- **Data source:** LINE for all Easy View programs
- **Disk:** HRD and **Disk Block No.:** 0 (fields appears only when **Data source** is changed to DAT or FILE)
- **File:** _____(field appears only when **Data source** is changed to FILE)
- **Clock:**
  - EXTERNAL for monitor and emulate DTE programs
  - INTERNAL for emulate DCE programs
- **Speed:** 9600
- **DTE Speed:** 9600 and **DCE Speed:** 9600 (fields appear only when **Clock** is INTERNAL SPLIT)
- **Use NRZI for SDLC:** NO

To edit self−contained parameters for a program, press [EDIT] again. The defaults (highlighted) and alternate selections for each parameter appear on the screen. See Figure 4-9.

> **NOTE:** Only those parameters which are applicable to a given program are available for editing. For example, **NRZI** can be modified for SDLC and SNA programs only and **Data source** can be changed from LINE for nonemulation programs only.

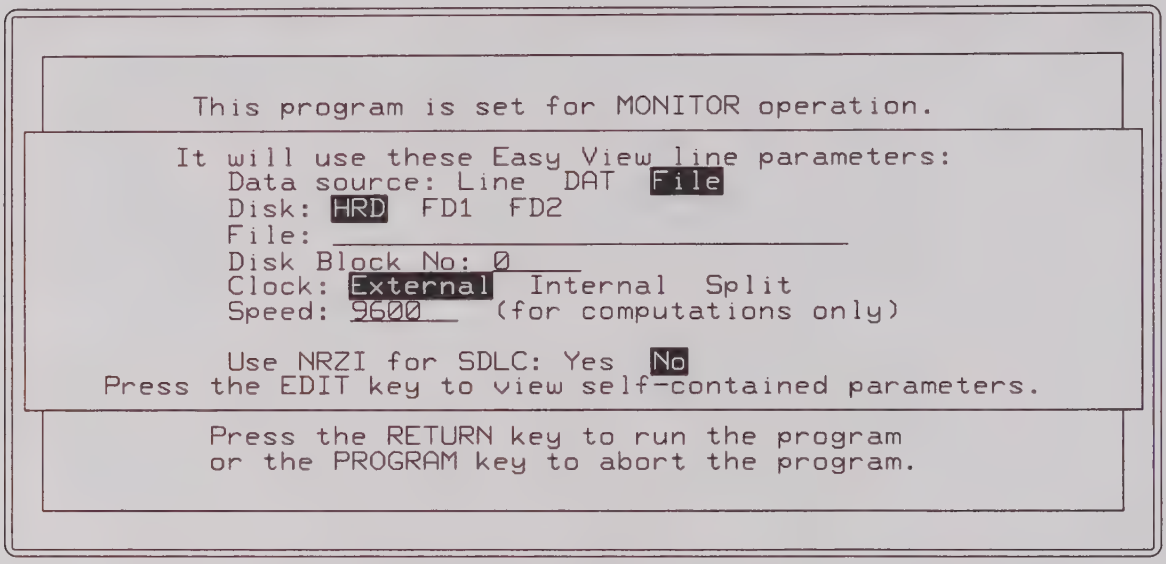


Figure 4-9 To modify selected line parameters for a program, choose from available selections displayed on the screen.

Use the → and ← keys to change the selection for a field. The selection is highlighted in reverse–video. Use the ↓ and ↑ keys to move between parameter fields. Press EDIT to toggle between the Easy View line parameters and the self–contained line parameters.

Once you have made your changes, press RETURN to run the program with the new Easy View line parameters. *The Easy View parameters are now the "default" line parameters for subsequent programs you run during the remainder of the current session (or until you change the parameters again).*

Assume, for example, that you changed the **Speed** from 9600 to 5600. The next time you select a program to run, it will be set to use 5600 as the line speed. See Figure 4-10.

> **NOTE:** If the Easy View parameters are set to use disk as the data source and you subsequently select an emulation program to run, the Easy View parameters are automatically updated so that the data source is changed back to line.

The most recent set of Easy View parameters are retained in memory for the rest of the working session. They are restored to the screen any time you access the Easy View parameter screen.

If you want to make permanent changes to the setup of a program, see Section 4.5(B).

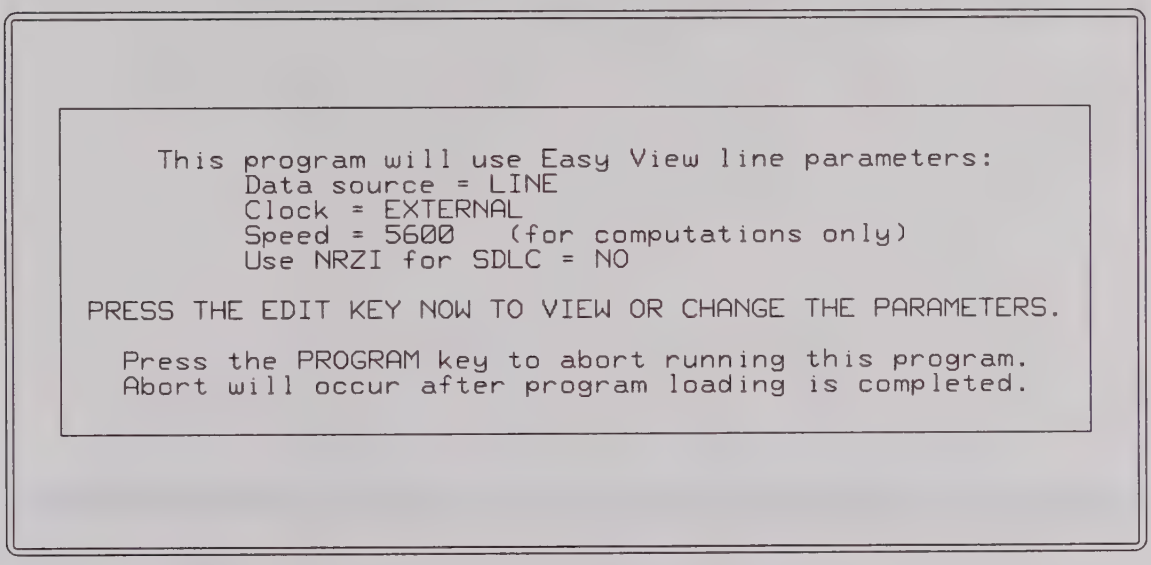


Figure 4-10 The program ready to run is set to use Easy View line parameters. The *Speed* has been changed from 9600 to 5600.

(D) Viewing Text Files

When you press ? (the / or SHIFT–/ keys) for help or select a menu item whose designated action is to present a text file for viewing (TUTORIAL or How To ...), Easy View attempts to load the proper file into memory.

If the file is not found, an error message is displayed. Press any key to return to the menu screen. If the file is located, it is presented for viewing. Help, Tutorial, and How To files are all presented in the same format.

1. *Format*. The format of the screen used to present text is shown in Figure 4-11. The top twenty screen lines form a scrolling window. The value at the right end of the bottom line is the percentage of the text file preceding the last character on the screen. An indication of the current location in the file appears at the left end of this line: (top), (more), or (end).

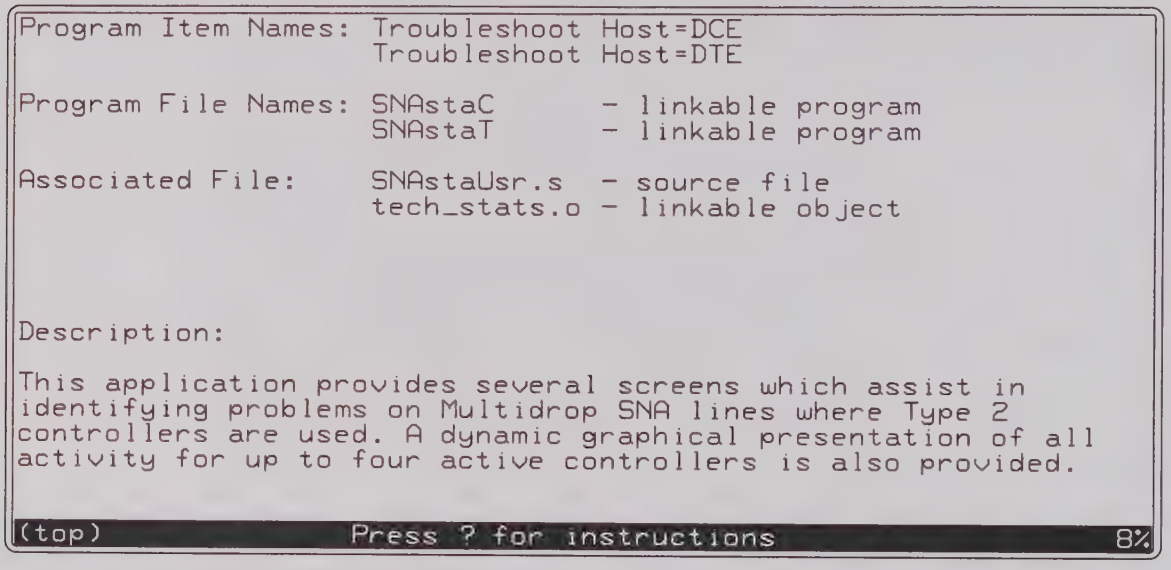


Figure 4-11 Information on SNA statistic program selected in Figure 4-5. To access this information, press ? while the selection bar is positioned over *Troubleshoot Host=DTE*.

2. *Keys*. Instructions for scrolling through information files are displayed when ? is pressed. These are shown in Figure 4-12.

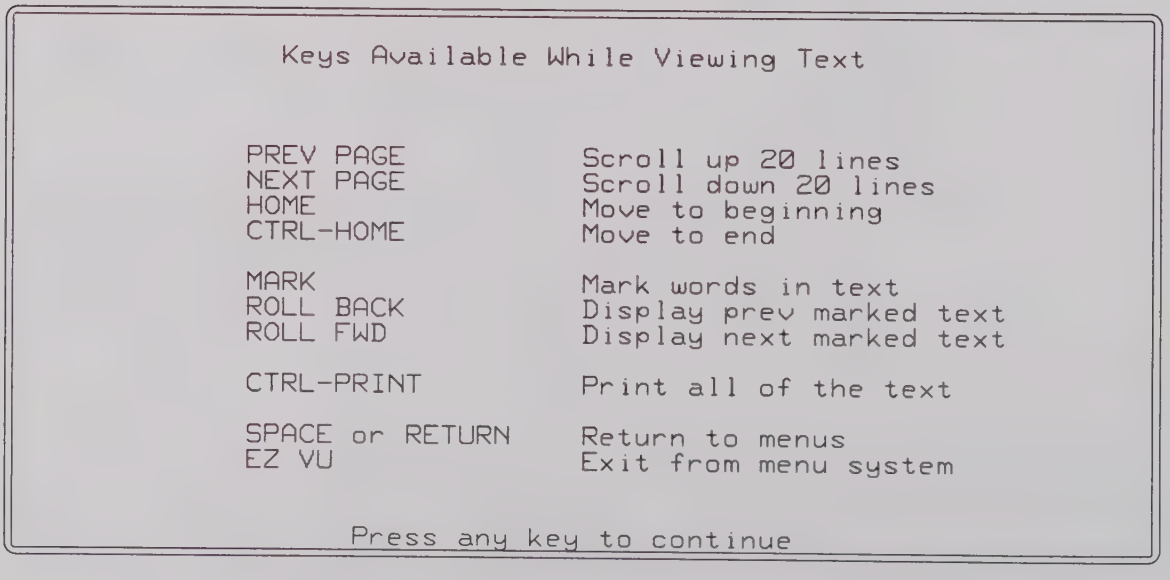


Figure 4-12 Keys available while viewing text files.

NOTE: If you want to print the contents of a text file, we suggest that you set the **Lines Per Page** field on the Printer Setup menu to 062. (See Section 16.) Each page of the printed output will contain a header and three of the INTERVIEW's text pages.

3. *Operation.* When you first enter a text viewing screen, the initial twenty lines of the file are presented. Refer again to Figure 4-11. Use the keys described above to scroll through the file. If Easy View menu information is being kept in memory and you subsequently leave Easy View (via the EZ VU key), the system records which portion of the file was being displayed at the time of exit. When you reenter Easy View, you are automatically returned to the same position in the text viewing screen as when you exited. Use this feature and the EZ VU key to flip between Program–mode screens and help information about the screens.

4. *Marking.* You may also "mark" all occurrences of a text string within any text file. The INTERVIEW searches for all *words* in which the specified string occurs and highlights, or marks, them in reverse video. The search is conducted throughout the entire text regardless of the cursor's current location within the file.

   For an example of marking, let's view the SNA tutorial. If you ran the Troubleshoot Host=DTE program, press PROGRAM to abort the program. Now press ← to return to the Menu of SNA Program Types, Tutorial(s) & How To's. Move the selection bar over the TUTORIAL menu entry and press RETURN. The screen shown in Figure 4-13 should be displayed.

```
   INTERVIEW SYSTEMS NETWORK ARCHITECTURE (SNA)  TUTORIAL

INTRODUCTION

The SYSTEMS NETWORK ARCHITECTURE (SNA), as it is defined and
implemented by IBM, provides the ability to connect many
different types of devices and programs into a complex network
that allows all of these different devices and programs to
communicate and interact.

This architecture defines the division of all of the network
functions into clearly defined layers.  These layers provide
many of the same functions as the Open Systems Interconnect
(OSI) seven-layered architecture defined by the International
Standards Organization (ISO) but the layers are not identical
and may not be inter-operable unless some form of gateway is
provided.

(top)              Press ? for instructions                  2%
```

Figure 4-13 First screen in the SNA tutorial.

Now press MARK. A pop–up window, illustrated in Figure 4-14, appears over the text.

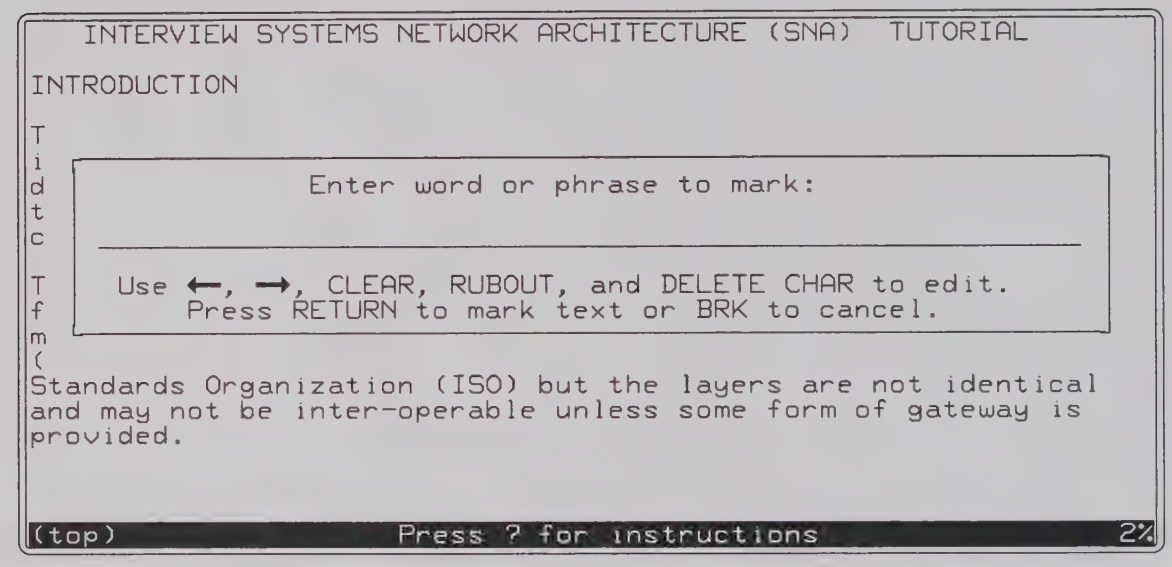


Figure 4-14 Press MARK to enter a search string for highlighting.

Type in the text string you want marked. The field is 56 characters long, including embedded spaces. (Multiple consecutive spaces are interpreted as a single space.) If you enter the string as lowercase characters, all words containing the string are marked regardless of case. If the search string contains any uppercase characters, only exact matches are highlighted.

Suppose you want to locate information about Physical Units (PUs) without scrolling through the entire SNA tutorial. Simply type the string "pu" and press RETURN. The screen containing the first occurrence of the marked text is automatically presented. See Figure 4-15. Marking is always expanded, if necessary, to highlight whole words (although blank spaces within a string are not highlighted). If no instance of the string is found, a message to that effect is displayed.

Look at the highlighted box at the very bottom of the screen. The prompt "MARKED ↓" tells you that more occurrences of marked text follow. The arrow direction—↓, ↑, or ↑↓—indicates the direction of additional marked text. Press ROLL FWD to move to the next screen containing the marked string. Use ROLL BACK to locate the previous screen with the highlighted text.

When you reach the last screen with an instance of the string, the directional arrow for marked text is ↑ and pressing ROLL FWD causes an alarm to sound.

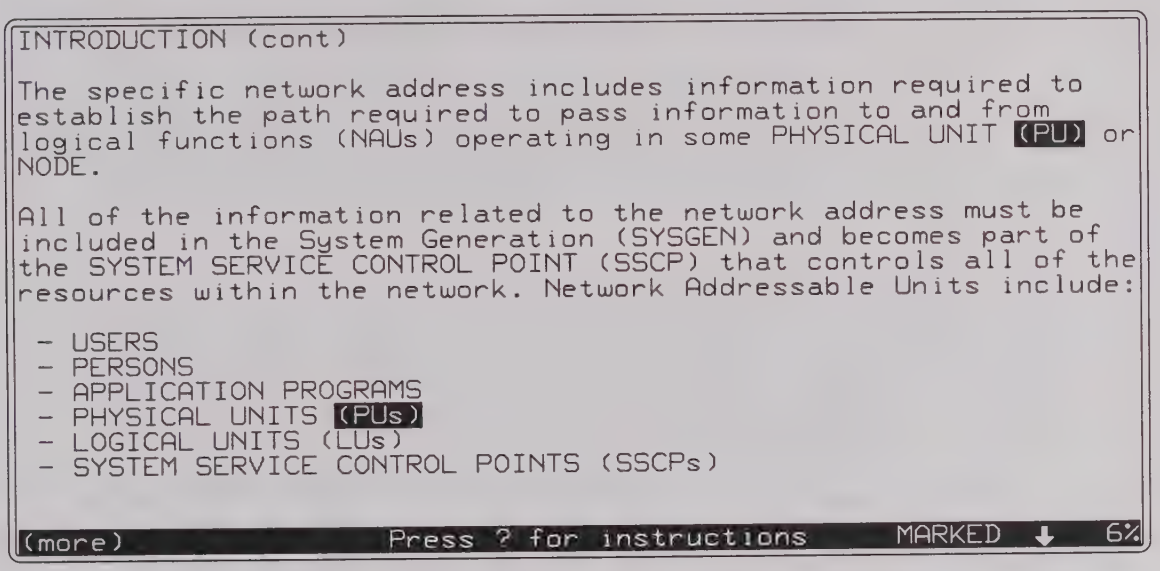


Figure 4-15 All words that contain the specified string—*pu*—are marked in reverse video. Press ROLL FWD and ROLL BACK to view additional screens with marked text.

Similarly, when the cursor is at the first screen containing the string, the directional arrow for marked text is ↓ and pressing ROLL BACK causes an alarm to sound.

To "unmark" the text file, press MARK again. Clear the text–entry line by pressing CLEAR. Press RETURN to execute. All highlighting is removed from the text.

4.5 Additional Easy View Functions

There are additional capabilities related to Easy View which are not essential to its basic operation.

(A) Changing Other Program Setup Parameters

The on–line editing feature allows you to modify selected line setup parameters. In some cases, you may want to change other default Setup menus for a selected program. These changes to the configuration of setup menus are made in Program mode.

1. Select the program from an Easy View menu and press RETURN to load the program. Or, use CTRL–LOAD to load and run the source version of the program—see Section 4.5(C).

2. Once the initial Run–mode screen is displayed, press PROGRAM and then EZ VU to enter Program mode.

3. Enter the appropriate setup screens and make any necessary changes.

4. Press [RUN]. The program will run using the new setup criteria. Since Easy View always reloads and runs the hard–disk version of a program, do not reenter Easy View to run the program or these changes will be overwritten by the default parameters stored with the original program.

(B) Saving New Setup Parameters

All AR–supplied application programs that you can load and run from Easy View are stored on the hard disk as linkable–program (LPGM) files. If you anticipate frequent use of a program with changes to the Setup menus, you may *replace* the original LPGM file with your newly configured version *using the same pathname*. Then, you can select and run the new version from Easy View. See Section 15 for information on the File Maintenance Save command and linkable–program files.

1. Select the program from an Easy View menu and press [CTRL]–[RETURN]. The full pathname of the program's LPGM file is displayed toward the bottom of the screen. This is the pathname you must use for your modified version.

2. Press [RETURN] to load the program.

3. Once the initial Run–mode screen is displayed, press [PROGRAM] and then [EZ VU] to enter Program mode.

4. Enter the appropriate setup screens and make any necessary changes.

5. Press [PROGRAM], then the FMAINT softkey to access the INTERVIEW's File Maintenance screen. When the program was loaded from Easy View, the File Maintenance screen was updated as though the program had been loaded from that screen. The current directory should be the one containing the program's file, and the program's filename should be the default entry in the **Name** field for most commands.

6. Select SAVE as the command. Confirm the entries in the fields on the screen. The **Type:** field should show LPRGRM. The entries in the **Drive** and **Name** fields should correspond to the name of the file displayed in step 1. Press [XEQ]. (If you need to modify the filename, remember that filenames *are* case–sensitive.)

> **NOTE:** You may also create multiple versions of a program to appear as separate entries in the same Easy View menu. See Section 20.

(C) Additional Menu Keys

There are two more key combinations that operate on Easy View menus:

[CTRL]–[RETURN] Displays the full pathname of the file associated with the selected menu entry.

[CTRL]–[LOAD] Active when the selection bar is positioned over a menu item whose designated action is to load and run a program. Press [CTRL]–[LOAD] to load and run the source version (*.s* extension) of the program—when it has been provided—instead of the linkable–program version. In Program mode, the spreadsheet program is available for viewing and modification.

4.6 Installing Easy View Updates

Easy View updates are not contained on the system or user disk of new system software. For information on when Easy View updates accompany your purchase of the latest system software, contact Customer Service.

The Easy View menus and programs are contained on several floppy diskettes. Follow these steps to install Easy View updates to the hard disk:

> *Caution: The following installation procedure will cause the existing Easy View menu structure and any existing program files with names that match new programs to be overwritten. This means that any changes to Easy View that you have previously made via the Easy View Maintenance installation procedure will be lost.*

1. Insert a diskette in Floppy Drive 1, the left–hand drive. (It does not matter in which order you duplicate the disks.)

2. If you are currently in the Easy View menus, press [EZ VU] to exit. Once in Program mode, if any screen other than the Program Menu appears, press [PROGRAM]. From the main Program Menu, press UTIL, D/MAINT to access the Disk Maintenance screen.

3. Now press the DUPDISK softkey to set up the Duplicate Disk Command. The **From Disk Number:** field on the screen should show FD1 ; select HRD in the **To Disk Number:** field.

4. Press [XEQ], then GOAHEAD to begin duplication of the first diskette. When the duplication is completed, the prompt "*Insert Next Disk –– Depress F1 Key To Continue*" will appear on the status line toward the top of the screen.

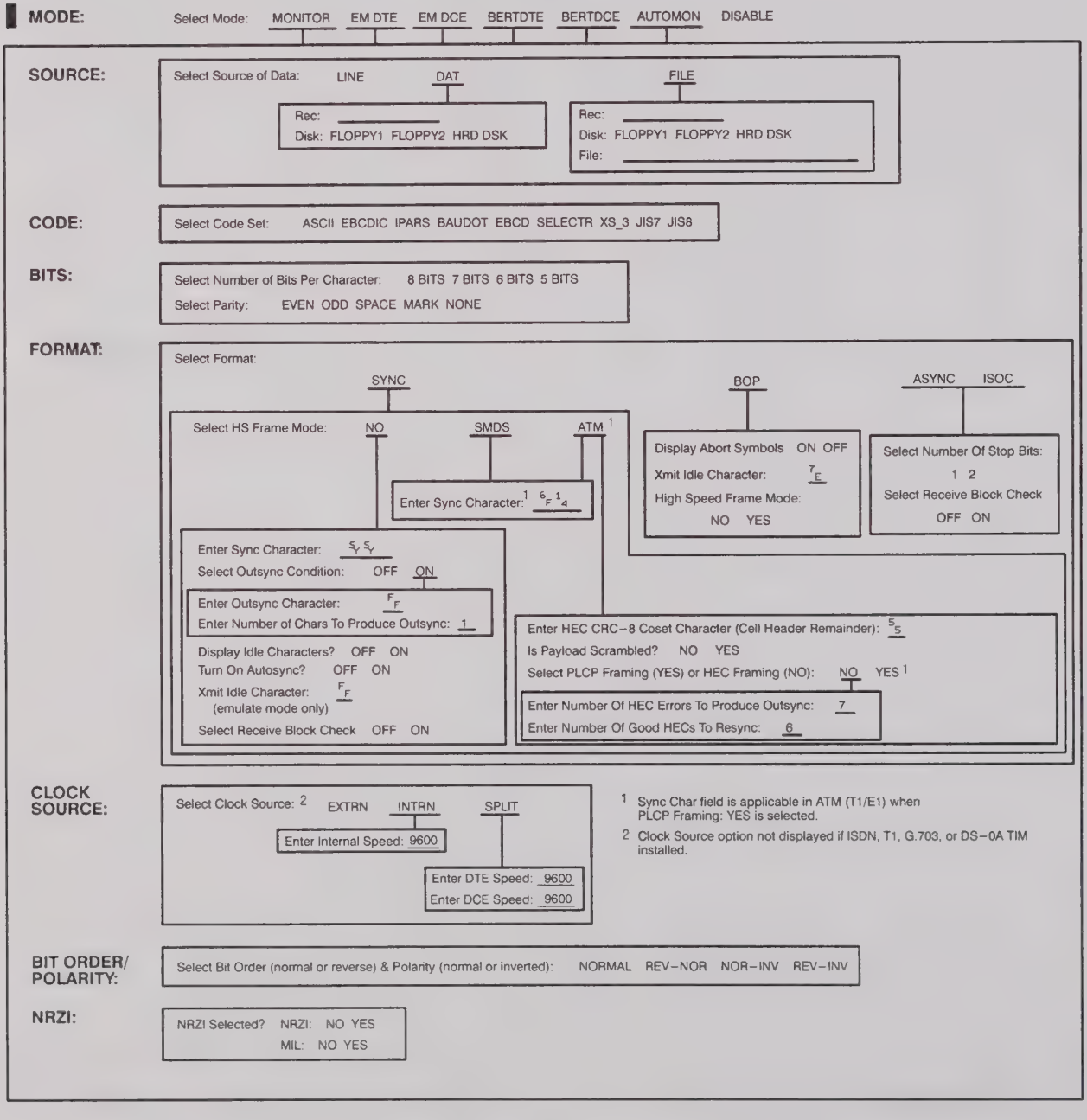
5. Remove the floppy disk from the drive and insert the next disk. Press GOAHEAD again. Repeat this process until all of the Easy View diskettes have been duplicated to the hard disk.

6. To see the new Easy View menus, power the unit off and then back on, or press CTRL – SHIFT – DELETE CHAR to reset the INTERVIEW.

7. If you want programs you had previously installed in the old Easy View menus to appear in the new Easy View menus, you must use the Easy View Maintenance program to reinstall them. See Section 20.

5 Line Setup

Line Setup



1 Sync Char field is applicable in ATM (T1/E1) when PLCP Framing: YES is selected.

2 Clock Source option not displayed if ISDN, T1, G.703, or DS–0A TIM installed.

Figure 5-1 Line Setup menu.

5 Line Setup

The Line Setup menu is the configuration menu for line parameters such as data format (synchronous or start–stop), code, clock, bit order, and bit polarity. The data may be received at a modular test interface, generated internally, or played back from disk. Figure 5-1 shows the fields and selections on this menu.

The Line Setup menu is on the left half of the Line Setup screen. Press PROGRAM, F1, F1 (or PROGRAM, RETURN, RETURN) to access the first field on this screen.

5.1 Mode

The first field on the Line Setup menu is **Mode**. In this field you select the type of operation you intend to perform with the INTERVIEW.

Available modes are Automonitor, Monitor, Emulate DTE, Emulate DCE, BERT DTE BERT DCE, and Disable. The Disable selection is only applicable when an ATM Interface Module (AIM) is installed in an ATM unit.

(A) Automonitor

Select AUTOMON (the default mode) if you are not certain of the line characteristics. When you press RUN, the unit will test samples of line data, autoconfigure, and begin to monitor. Once the unit is configured, change the mode to MONITOR. This is to prevent the unit from autoconfiguring every time you go into Run mode.

> **NOTE**: If AUTOMON is selected as the mode, any installed ATM Interface Module will not be recognized on the Setup Menu screen.

To use Automonitor, you must also select **Source:** LINE. **Source:** DISK does not provide data samples for the unit to test. In this instance, it cannot autoconfigure.

(B) Monitor

Select **Mode:** MONITOR if you intend simply to monitor data, and you wish to enter the configuration parameters yourself. The INTERVIEW is a passive monitor (that is, it does not regenerate signals). Once the unit's DTE and DCE connectors are attached to the data line, the unit will not interfere with data transmissions when it is in Automonitor or Monitor mode. Select **Mode:** MONITOR also when you want to play back data from a disk.

NOTE: Before data can be played back or the line monitored, all transmit actions must be cleared from the trigger menus, and all SEND actions and RECEIVE or RCV conditions must be deleted from the Protocol Spreadsheet. These conditions and actions are for emulate mode only.

(C) Emulate DTE

Make this selection for interactive testing of a modem or testing of a DTE *across* modems. Break the data line from the DTE and connect the modem to the TO DCE interface on the interface module at the rear of the INTERVIEW.

Note on synchronous and BOP operation: The INTERVIEW can transmit using internal clock (SCTE) when it is emulating a DTE; but in order to display its own transmit data as well as receive data, it must be connected to a modem or other DCE device that is providing SCT and SCR clock.

If you are emulating DTE and external clock is not available for synchronous or BOP operation, select **Clock Source:** INTERNAL and patch SCTE to SCT.

(D) Emulate DCE

The EMULATE DCE mode is for direct, local testing of a DTE (such as a terminal, PAD, or front end). Break the data line from the modem and connect the DTE to the TO DTE connector on the interface module at the rear of the unit.

(E) BERT DTE

In BERT DTE mode you will run the BERT test that is configured on the BERT Setup screen (Section 11), using the TD lead to transmit and RD to receive and analyze.

(F) BERT DCE

In BERT DCE mode you will run the BERT test that is configured on the BERT Setup screen (Section 11), using the RD lead to transmit and the TD lead to receive and analyze.

NOTE: The BERT softkey and description text line on the Setup Menu will not appear *unless* one of the BERT modes on this screen is selected.

If either BERTDTE or BERTDCE *is* selected, the AIM softkey and description text line on the Setup Menu will not appear.

(G) Disable

Use the DISABLE mode *only* when testing Broadband ATM data *exclusively*; it "turns off" the WAN test processors and allocates that memory to the ATM functions and applications (see Figure 5-2). This mode also disables access to the Interface Control screen, BCC Control screen, and Front–end Buffer Setup screens: their softkeys and text lines on the Setup Menu disappear. Change the mode to return the standard Line Setup features.

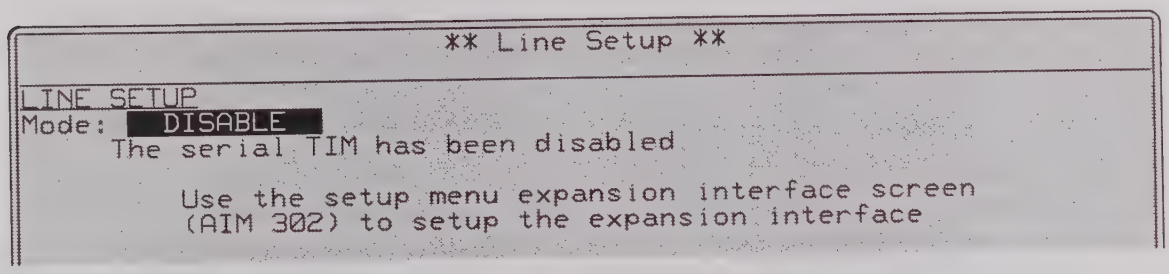


Figure 5-2 Line Setup menu disabled.

5.2 Source

This selection field determines the source of the data to be monitored or tested.

LINE is the default selection. In Monitor or Automonitor mode when LINE is selected, the INTERVIEW will monitor data received at either of the line–interface connectors on the interface module at the rear of the unit.

In Emulate DTE mode when LINE is selected, the INTERVIEW will transmit on pin 2 (TD) and expect to receive data on pin 3 (RD). In Emulate DCE mode, the unit will transmit on pin 3 (RD) and monitor pin 2 (TD).

Part of any disk may be set aside for the recording of live data. Line data may be recorded directly to disk, recorded to RAM and then transferred to disk, or captured in the screen buffer (RAM) and transferred to disk. To record to disk or RAM, use the fields of the Record Setup menu on the Line Setup screen (see Section 7, Record Setup). Capture in the screen buffer (character RAM) is discussed in Section 26.9.

For data to be available for playback, it must reside on a disk. Therefore, data stored in bit–image RAM or screen buffer RAM must be transferred to disk using the Data Transfer command. See Section 14.4(C).

On the Line Setup menu, the **Source:** DAT selection allows data that has been recorded to the data–acquisition track to be played back for testing and display. Selecting DAT brings up two new fields: **Rec** and **Disk**.

The **Source:** FILE selection allows data that has been recorded to a file (rather than data–acquisition tracks) to be played back for testing and display. Selecting FILE brings up three new fields: **Rec, Disk,** and **File.**

For either DAT or FILE, enter the starting block number in the **Rec** field. Valid entries are 0 to 9999999999. If you do not enter a value, the value will default to zero. Block 0 means that playback will start from the beginning of the data source, regardless of the actual block number, just as any block number beyond the end of the data sample through 9999999999 will take you to the end of the data source. All other entries represent actual block numbers.

NOTE: It is not necessary to enter leading zeroes in this field. For example, you may enter 10 instead of 0000000010.

Choose from FD1 (FLOPPY1), FD2 (FLOPPY2), and HRD (HRD DSK) in the **Disk** field when playing back data from either Data Acquisition Tracks or a file.

When you choose **Source:** FILE , you also need to identify the data file in the **File** field. Enter the full pathname of the file, up to 29 characters excluding the disk designation (which is read from the **Disk** field). An example of a filename entry for this field is */usr/Data1291*.

NOTE: Data playback from a file is slower than from the data acquisition tracks.

NOTE: Filenames in the INTERVIEW file system are *case–sensitive*. You must enter the name exactly as it appears in the directory listing.

When DAT or FILE is the source selection, a record/playback field (on the top line in the status area) on Run–mode screens will indicate whether playback is in progress or has been suspended. If playback is ongoing, a "P" appears next to the block–number field. If you suspend playback by pressing RECORD or using a program action, "S" will be displayed. The record/playback field will be blank if a disk is not present in the selected drive or when the end of the data—acquisition tracks or file is reached. This field will also be blank if you enter a starting block number on the Line Setup menu that a) precedes the block number at which data actually begins, or b) exceeds the block number at which data actually ends. Change your entry to zero.

NOTE: An "R" in the record/playback field indicates that line–data recording is in progress. See Section 7.

5.3 Code

The standard **Code** selections are EBCDIC , ASCII , EBCD , XS_3 , IPARS , BAUDOT , SELECTRIC , JIS7 , and JIS8 . Default is EBCDIC .

Appendix D1 gives keyboard–translation tables for all the codes that are standard in the INTERVIEW. These tables indicate the hex byte that is transmitted (or searched for) and the actual character displayed in Run mode as a result of a given keystroke (unshifted, shifted, or control) made while the cursor is in a data–entry field.

Note the difference between Run–mode display and Program–mode display of keyboard input. In Program mode, a particular keystroke (CTRL pressed in conjunction with ▫, for example) will always produce the same character ("~") in a data–entry field. In Run

mode, a particular code–translate chart will be consulted and the character (" ~ " in our example) may not belong to the code set and therefore may be missing on the chart. Such a character cannot be trapped, transmitted, or displayed in Run mode: it is "untranslatable" (see note, Appendix D1).

Appendix D2 gives an input–to–display translation table for all of the standard codes. Next to each hexadecimal value (input) is the ASCII–keyboard character or control character (display) in each of the various codes. "Input" refers to bytes, both received and transmitted, in hexadecimal notation before they are converted to characters and displayed on the monitor.

5.4 Bits

Identify the number of information bits for the selected code in the **Bits** field. Do not include the parity bit. Options are 8, 7, 6, and 5.

5.5 Parity

In the **Parity** field select the type of parity used in the system being tested. The choices are NONE, EVEN, ODD, SPACE, and MARK. The parity bit is additional to the information bits.

A data character received with a parity error will appear on the display with a bar through it.

> **NOTE:** 8–bit odd or 8–bit even parity is a functional choice for ASYNC or ISOC format. Eight bits plus mark or space parity is not a valid selection for any format.

5.6 Format

The **Format** selection field allows you to program the INTERVIEW correctly for the protocol to be monitored. The selection in the **Format** field determines many of the selection subfields that follow.

(A) Sync Selections

Selecting SYNC for **Format** results in the group of subfields shown in Figure 5-3. Choose SYNC for Bisync format.

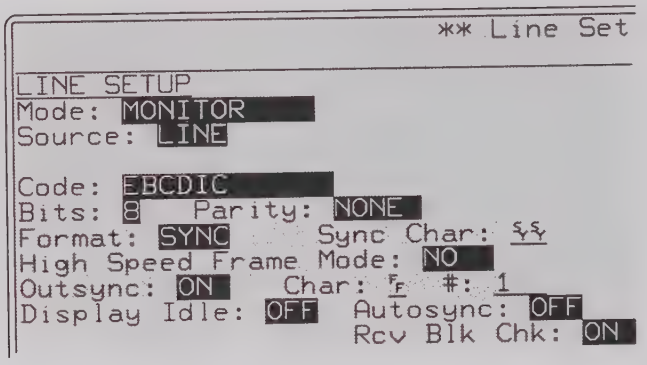


Figure 5-3 There are eight subfields under *Format: SYNC*.

1. *Sync characters.* **Sync Char** is a data–entry field that determines the synchronization pattern for synchronous data protocols. When the receivers are looking for synchronization, every occurrence of the sync pattern results in a S (sync) symbol on the data display. Refer to **Outsync** and **Autosync** (subsections 3. and 5. below) for the conditions under which the receivers *look for sync*.

   The synchronization characters default to SYN SYN. Adjustments for your parity selection are automatic. For example, SYN in odd–parity ASCII is 16 but in even–parity ASCII the INTERVIEW converts SYN to 96. For most cases, therefore, you will not have to make any entry in the **Sync Char** field.

   IPARS and XS_3 are exceptions in that the control character SYN does not occur on their translation tables. If SYN is entered as a sync character for these two codes, it is treated like any other untranslatable character and converted to SPACE or NULL, inappropriate for syncing.

   For IPARS, the hexadecimal value 3F 3E should be entered manually in the **Sync Char** field (see Figure 5-4). In the case of XS_3, the correct sync pattern is 3B 3B.

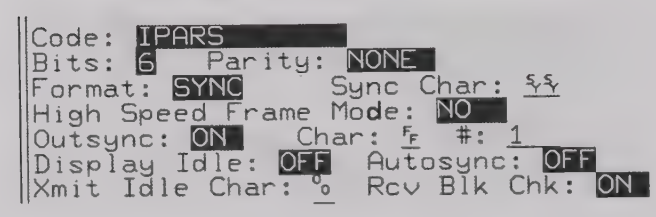


Figure 5-4 Different codes have different sync parameters.

The correct synchronization patterns for the various standard codes in the INTERVIEW are listed in Table 5-1. Any other one– or two–character sequence may be entered, using alphanumeric keys, control characters, or hexadecimals.

The character entered will be adjusted for parity unless **Parity:** NONE has been selected or unless the entry has been made in hexadecimal. *For hexadecimal entries no parity adjustment is made. You must adjust hexadecimal entries yourself to take parity into account.* (Do not adjust parity for the default hexadecimal sync characters in IPARS or XS−3.)

To enter a one−character synchronization pattern, position the cursor in the **Sync Char** field and depress CLEAR. This will clear the field. Enter the desired character.

(a) *SMDS Framing Mode.* When **HS Frame Mode:** SMDS is selected, the synchronization pattern is $^{6}_{F}\,^{1}_{4}$.

(b) *ATM Framing Mode.* The **Sync Char** field entry is not relevant in ATM Framing Mode *unless* **PLCP Framing:** YES is also selected. In this case, the synchronization pattern is also $^{6}_{F}\,^{1}_{4}$ as in SMDS Framing Mode; otherwise, the **Sync Char** field entry is not relevant in ATM Framing Mode (see subsection 2.(c), below).

Table 5-1
Synchronization Parameters for Standard Codes

Code	Info Bits	Parity	SY1 SY2
EBCDIC	8	None	$^{3}_{2}\,^{3}_{2}$
ASCII	7	Odd	$^{1}_{6}\,^{1}_{6}$
		(Space	$^{1}_{6}\,^{1}_{6}$)
		(Even or Mark	$^{9}_{6}\,^{9}_{6}$)
EBCD	6	Odd	$^{3}_{D}\,^{3}_{D}$
		(Space	$^{3}_{D}\,^{3}_{D}$)
		(Even or Mark	$^{7}_{D}\,^{7}_{D}$)
XS−3	6	Odd	$^{3}_{5}\,^{3}_{5}$
IPARS	6	None	$^{3}_{F}\,^{3}_{E}$
SELECTRIC	6	Odd	−−

2. *HS Frame Mode.* Some INTERVIEW 8000 Series units now support applications for monitoring and emulation on the SMDS SNI (Switched Multimegabit Data Service Subscriber Network Interface) and in ATM UNI (Asynchronous Transfer Mode User Network Interface). To enter these modes, select either SMDS or ATM in the **HS Frame Mode** field; the field defaults to NO.

   SMDS. No additional hardware is required to run applications for monitoring and emulation on the SMDS (SNI) at T1 and E1 data rates. AR Test Systems applications currently available for SMDS SNI include:

   SMDS SNI Trace and Statistics [OPT−951−133−1]
   SMDS SNI Emulation with Trace and Statistics [OPT−951−197−1]

NOTE: AR Test Systems also has applications available for SMDS DXI while in BOP High Speed Frame Mode—see Subsection 10.4(A)2. These include:

SMDS DXI Trace and Statistics [OPT−951−199−1]
SMDS DXI Emulation with Trace and Statistics [OPT−951−189−1]

The SMDS SNI for T1 is defined in Bellcore document TR−TSV−000772 as a public service offering. The SNI is based on 53-byte cells and uses a PLCP (Physical Layer Convergence Protocol) to map cells onto the T1 framed data streams. Note that the SMDS specifications require that all octets are transmitted *most significant bit* first.

To enter this mode, select **HS Frame Mode:** SMDS . In SMDS HS Frame Mode the **Outsync, Char, #, Display Idle, Autosync,** and **Rcv Blk Chk** fields disappear (see Figure 5-5) as they are not applicable; however, enter sync pattern $^{6}_{F}\,^{1}_{4}$ in the **Sync Char** field. In this mode the system software passes groups of cells up to the application with each IL message buffer; this is a useful tool for the advanced C−programmer. At this time, application programs for SMDS are available; however, there is no direct system software support for SNI transmission.

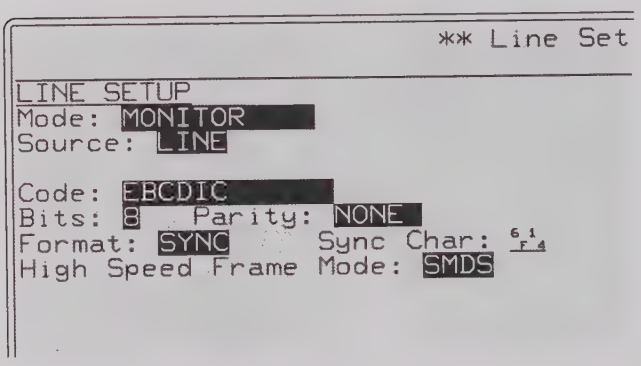


Figure 5-5 In SMDS High Speed Frame Mode, the **Outsync, Char, #, Display Idle, Autosync,** and **Rcv Blk Chk** fields disappear; note sync character pattern is $^{6}_{F}\,^{1}_{4}$.

ATM. No additional hardware is required to run applications for monitoring and emulation in ATM at T1 and E1 data rates. At present, there are five application programs for ATM for the INTERVIEW 8000 Series units:

Trace and Statistics: Asynchronous Transfer Mode (T1/E1) [OPT−951−232−1],
Emulation with Trace and Statistics: Asynchronous Transfer Mode (T1/E1) [OPT−951−233−1], and
Asynchronous Transfer Mode Cell Loss and Delay Test (T1/E1)[OPT−951−237−1]
Emulation with Trace and Statistics: Asynchronous Transfer Mode DXI (T1/E1) [OPT−951−309−1]
Trace and Statistics: Asynchronous Transfer Mode DXI (T1/E1) [OPT−951−310−1]

As Telenex is continually developing new products and programs, contact Customer Service for updated release information.

Like SMDS SNI, ATM is based on 53−byte cells; an INTERVIEW 8800 *TURBO* unit verifies the cell headers and displays the cells' payload in hexadecimal format. The unit handles either PLCP framing or standard ATM HEC framing. ATM for T1 is defined in the ATM Forum DS1 ATM UNI (draft) Specification and the ITU (draft) documentation I.432 and G.804.

To enter ATM Framing Mode, select **HS Frame Mode:** ATM. In this mode the **Outsync, Char, #, Display Idle, Autosync,** and **Rcv Blk Chk** fields disappear as they are not relevant in ATM (see Figure 5-6). Five new fields appear: **HEC CRC−8 Coset, Payload Scrambled, PLCP Framing, Outsync HECs,** and **Resync HECs.** Note the sync character is not applicable if **PLCP Framing:** NO is selected, even though the field remains displayed; when selecting PLCP framing mode, enter sync pattern $^{6}_{F}\,^{1}_{4}$ as in SMDS Frame Mode.

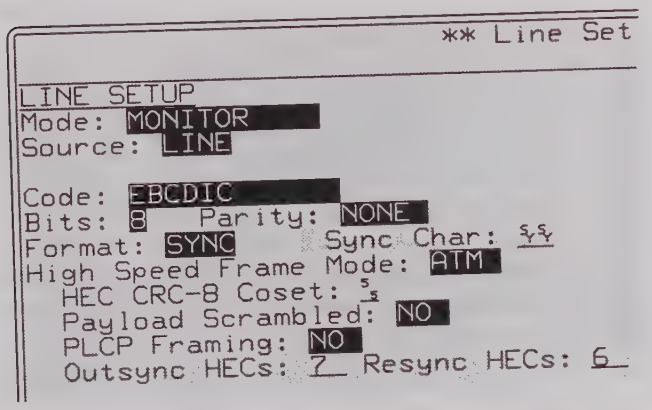


Figure 5-6 In ATM High Speed Frame Mode, the **Outsync, Char, #, Display Idle, Autosync,** and **Rcv Blk Chk** fields are replaced by five new fields: **HEC CRC−8 Coset, Payload Scrambled, PLCP Framing, Outsync HECs,** and **Resync HECs.**

(a) *HEC CRC−8 Coset.* This field represents the HEC (Header Error Control) CRC−8 Coset character. Enter any hexadecimal number in this field; $^{5}_{5}$ is the recommended and the default entry. This entry is *XOR*ed (exclusive/or) with the cell header remainder, the value derived from a polynomial ($X^8 + X^2 + X + 1$) operation on the cell header, the first four characters of the cell. The result is the fifth cell character.

(b) *Payload Scrambled.* Generally when monitoring T1 data, the **Payload Scrambled** field entry should be NO, as the incoming data is usually in standard format. When monitoring E1 data, the recommended field entry would be YES, as the line data is generally scrambled; this selection enables descrambling of the data by the INTERVIEW 8000 Series unit. This descrambling will minimize the possibility of false cell synchronization.

The scramble/descramble operation occurs logically in a 43–bit shift register as each data bit is *XOR*ed as shown in Figure 5-7. Please note that the INTERVIEW 8000 Series units only *descramble* the data; you as the operator must know whether the data is scrambled and so requires this operation.

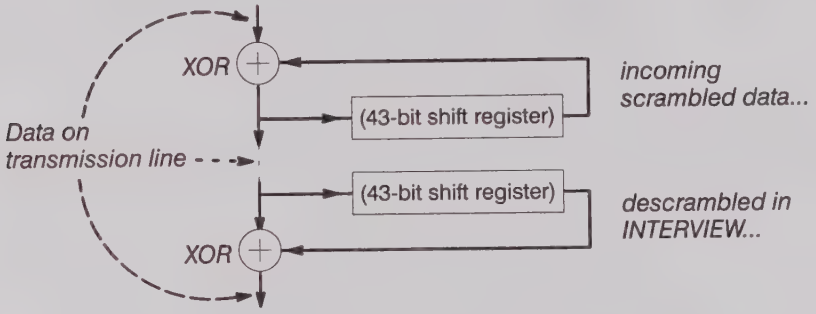


Figure 5-7 When **Payload Scrambled:** YES is selected, the incoming data will already have been XORed and logically scrambled; then the INTERVIEW 8000 Series unit descrambles it and puts it out on the line.

(c) *PLCP Framing.* The default selection for this field is NO which allows for standard ATM HEC framing. If PLCP framing is desired, select **PLCP Framing:** YES. This will enable decoding of a 576–byte superframe (ten 57–byte cells plus six trailer bytes) to follow the PLCP frame format; use sync pattern $^{6}F^{1}4$ with PLCP Framing. Note that the next two fields, **Outsync HECs** and **Resync HECs**, disappear when PLCP framing is enabled as these two fields are applicable *only* in HEC framing mode.

(d) *Outsync HECs.* In this field the user will enter the number of contiguous HEC errors which must occur in order to lose synchronization (to be *out of sync*). The **Outsync HECs** field defaults to *7*, but it will accept entries from *3* through *16*. This field is valid only in HEC framing mode and will disappear if **PLCP Framing:** YES is selected.

(e) *Resync HECs.* When the unit is in Cell Delineation Loss, the **Resync HECs** field allows the user to specify the number of contiguous good Header Error Control characters on which to resynchronize. This field defaults to *6*, but acceptable entries include *2* through *16*. This field is valid only in HEC framing mode and will disappear if **PLCP Framing:** YES is selected.

3. *Outsync.* When the unit enters Run mode in synchronous format, the receivers are always *out of sync, looking for sync*. After initial synchronization, the receivers stop looking for sync. **Outsync:** ON means that the receivers can *return* to the original outsync, looking–for–sync condition as soon as a particular outsync pattern is encountered. Unless this pattern is encountered (or unless **Autosync:** ON is selected), synchronization can occur only once during Run mode.

The default **Outsync** sequence is one ▨ (pad) or hex FF character. Any character may be entered in the **Char** field. The **#** field is a decimal field. It allows you to specify how many times the character must occur consecutively (from one to 255 times) before outsync will take effect. In transparent text, there is a possibility that legitimate data FF will occur. Also, the block−check character may occasionally mimic the outsync character. Normally these problems are solved by turning outsync off (and autosync on) or by increasing the number of outsync characters to 2.

Note these three fields will disappear when **HS Frame Mode:** SMDS or ATM is selected.

4. *Display idle.* **Display Idle:** OFF prevents display and buffer−storage of idle characters from the time the receivers go out of synchronization until they see the synchronization pattern again. If you select ON, the idle characters will be displayed on the monitor and also saved in the data buffer.

   Note this field will disappear when **HS Frame Mode:** SMDS or ATM is selected.

5. *Autosync.* When **Autosync** is ON, the receivers will resynchronize every time they see the selected synchronization pattern, *even if they are in sync already.* When autosync is enabled, the logic constantly tests for the one− or two−character pattern on a bit−by−bit basis. When a match is found, it becomes the new reference point for character framing.

   This is a useful selection when data does not have a unique outsync character (as when a transmitter idles the sync character, for example). Autosync is also useful where one block of data follows another by less than a full character interval. Autosync will detect the synchronization pattern and adjust to it even though it is skewed in relation to the previous sync.

   As long as autosync is off, there is no danger of accidental syncing in the middle of a block of data, since receivers that are already in sync do not look for sync. While autosync is on, accidental syncing may happen when a sequence of bits in the data mimics the sync pattern. You may reduce the chance of random syncing in the middle of a block of data by always using a two−character synchronization pattern whenever autosync is turned on.

   In normal synchronous operation, outsync is on and autosync is off. Once the receivers go out of sync, the receiver logic tests for the one− or two−character sync pattern one bit at a time, just as when autosync is on. Enabling autosync means that the bit−by−bit search is conducted all the time, even when the receivers are already in sync.

   Note this field will disappear when **HS Frame Mode:** SMDS or ATM is selected.

6. *Block−check overlays.* **Rcv Blk Chk** is a subfield of **Format:** SYNC, ASYNC, and ISOC. This field is discussed in Section 10, Block Checking. Note this field will disappear when **HS Frame Mode:** SMDS or ATM is selected.

7. *Transmit idle character.* The **Xmit Idle Char** field is valid for both SYNC and BOP formats. It allows you to specify what idle–line condition will be applied by the INTERVIEW in Emulate modes. Alphanumeric, control, or hexadecimal characters may be entered in this field. The default idle–line condition for SYNC format is steady mark (hex $^{F}_{F}$). Note this field will disappear when **HS Frame Mode:** SMDS or ATM is selected.

(B) Bit–Oriented Protocols

Format: BOP selects bit–oriented protocols that use $^{7}_{E}$ framing and zero–insertion. Examples are X.25, X.75, and SDLC.

In BOP protocols, the synchronization flag pattern, conditions for outsync, and block–check calculation are always defined. Any fields on the Line Setup menu that pertain to synchronization and block–checking disappear when BOP is the **Format** selection. The subfields under **Format:** BOP are **Display Abort, Xmit Idle Char,** and **High Speed Frame Mode.** See Figure 5-8.

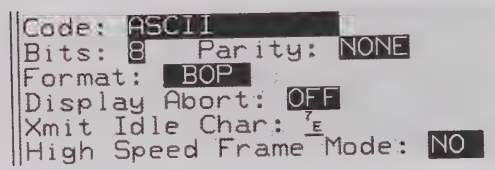


Figure 5-8 Synchronization and frame–check parameters are automatic (and therefore not selectable) under *Format: BOP.*

1. *Display abort.* The seven 1–bits in a row that signal an "Abort" to BOP receivers can occur during $^{7}_{E}$ idle as well as in the middle of a frame. Since these idle aborts are not aborted frames but merely indicate missing zeroes in the $^{7}_{E}$ –idle bitstream, you may want to suppress the display of these "glitches" by selecting **Display Abort:** OFF. This is the default selection. **Display Abort:** OFF does not pertain to aborted *frames*, which will be displayed in any case.

2. *Transmit idle character.* This field is valid for both SYNC and BOP formats. See under SYNC, above. The default idle character for BOP format is the $^{7}_{E}$ flag. Press [HEX], [7], [E], (not [FLAG]) to enter this idle character. Only $^{7}_{E}$ or $^{F}_{F}$ idle is valid for BOP.

3. *High speed frame mode.* Frame mode is a high–speed capture method in which the *frame* is the lowest entity. Received data is read directly into the IL buffers; this process allows full data capture at high speeds. Choose **High Speed Frame Mode:** YES to operate in Frame mode, such as when running on the SMDS DXI Switched Multimegabit Data Service Data Exchange Interface). The default is NO.

When you choose **High Speed Frame Mode:** YES, the additional fields for **Display Mode:** DATA are not applicable; the **Type**, **Enhance**, and **Suppress** fields, therefore, do not appear on the Display Setup screen when you choose to operate in Frame mode.

In addition, various C events, variables, and routines associated with character capture and character display (e.g., *rcvd_char–rd*, *rcv_buffer_full*, *td_modifier*, *rd_buffer_off*, *td_il_buffer*, *evar_rcv_char*, *fevar_abort*, *ctl_enhance_td*) cannot be used while operating in Frame mode. Refer to the Volume 2 Index to locate specific information on these events, variables, and routines; a complete listing is found in Section 2.10.

CAUTION: High–speed Frame mode operates at speeds above the AM band. RF cabling considerations must be taken when you operate in Frame mode.

(C) Start–Stop Format

For start–stop data, choose **Format:** ASYNC or ISOC. Then you may choose either 1 or 2 **Stop Bits**, as shown in Figure 5-9. ASYNC is the correct **Format** selection for data that has start/stop–bit framing if both digital devices rely on internal clock. If clock is provided on the interface for one of the devices (isochronous operation), select **Format:** ISOC.

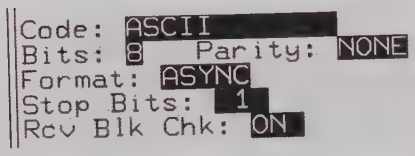


Figure 5-9 Async receivers go out of sync when a selectable number of stop bits are received.

5.7 Clock Source

Clock Source and **Speed** apply to data in any format. Selections for **Clock Source** are EXTERNAL, INTERNAL, and INTERNAL SPLIT. The **Clock Source** field is not displayed if an ISDN, T1, G.703, or DS–0A TIM is installed, as these TIMs are designed to provide the appropriate clock signals when in use.

If **Format:** ASYNC has been selected, the clock source *must be internal* (or internal–split). Async means that there is no clock on the interface leads, and the INTERVIEW will not look for external clock nor recognize the clock signals if they are there.

Note on synchronous and BOP operation: The INTERVIEW can transmit using internal clock (SCTE) when it is emulating a DTE; but in order to display its own transmit data as well as receive data, it must be connected to a modem or other DCE device that is providing SCT and SCR clock.

If you are emulating DTE and external clock is not available for synchronous or BOP operation, select **Clock Source:** INTERNAL and patch SCTE to SCT.

INTERNAL SPLIT allows you to monitor or emulate in *asynchronous* systems that have different baud rates for Receive Data and Transmit Data, and to emulate a modem on dual–speed, *synchronous* lines. When INTERNAL SPLIT rotates into the selection field, speed settings for both DTE and DCE appear below it. See Figure 5-10.

If you are monitoring or emulating a DTE on a dual–speed, synchronous line, the proper clock–source selection is EXTERNAL, since the modem is already supplying both clocks.

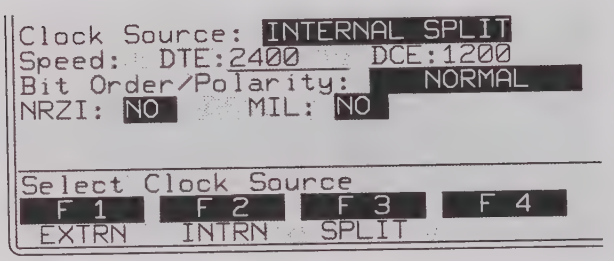


Figure 5-10 Internal clock may be split into DTE and DCE speed settings.

5.8 Bit–Order/Polarity

Bit Order/Polarity: NORMAL means that both the bit order and the bit polarity are "normal." REVERSE–NORM means that the bit order is reversed while the polarity is normal. Other selections are NORM–INVERT and REV–INVERT.

Normal bit order means that when a character is encoded into binary for transmission (or decoded from binary for display), the first bit transmitted (or received) will correspond to the rightmost bit in the INTERVIEW's hexadecimal value for that character and code. The code–conversion for every hexadecimal value is given in Appendix D2, Hex–to–Display Translation Tables.

When reverse bit order is selected, the hexadecimal display for each byte does not change; but now the first bit to be transmitted and received will be the leftmost bit in the binary version of the hexadecimal. (Leftmost, that is, except for the parity bit. For obvious reasons, the parity bit is always the last bit transmitted.) Reverse bit order is appropriate for certain standard codes such as IPARS and EBCD (which then becomes REV EBCD).

Usually the rightmost bit for the hex bytes on a translation table is the least−significant bit, but not always: the right−hand bit is the most−significant bit in two older IBM code sets, EBCD and SELECTRIC. But whatever the code set, normal bit order establishes a correspondence between the rightmost bit on the hexadecimal table and the first bit transmitted or received.

For normal bit polarity, the INTERVIEW's RS−232/V.24 transmitters define logic 1 as more negative than −3 V and logic 0 as more positive than +3 V. When you invert the bit polarity by selecting **Bit Order/Polarity:** NORM−INVERT or REV−INVERT, logic 1 becomes +3 V and logic 0 switches to −3 V. Logical inversion changes the voltages for bits in data characters (⅛, the ASCII sync character, becomes −−−+−++− instead of +++−+−−+) but the inversion does not affect voltages for start/stop−bit framing and mark idle. The start bit still signals the transition from negative to positive voltage. Stop bit and idle are still negative voltages, but now they are read as logic 0 instead of logic 1.

Note that the display LEDs above the INTERVIEW monitor always shine red when the voltage level is +3 V or greater. This is true for data leads as well as control leads, regardless of the data logic (polarity).

5.9 NRZI

When **NRZI:** YES is selected, INTERVIEW transmitters and receivers use the NonReturn to Zero (Inverted) mode of signal sense. In this mode, logic 0 is defined not as a plus−voltage threshold, but rather as any transition from plus to minus or from minus to plus. Logic 1 is defined as an absence of transition.

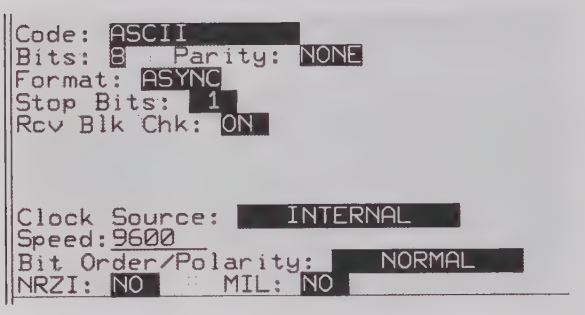
> **NOTE:** Since the hardware that accomplishes NRZI formatting is not on the playback path for disk data, **NRZI:** YES must be selected when the original line data enters the FEB. The **NRZI** selection will have no effect during playback, whether the data was recorded in character or bit−image format.

5.10 MIL

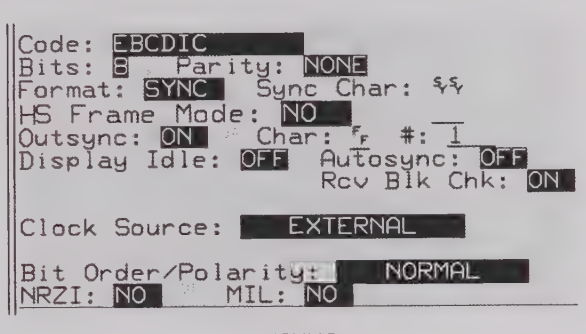
MIL: YES designates MIL−188 operation, which inverts all signals on the data leads (including idle). In MIL operation there is no need to invert bit polarity. Note that MIL operation does not convert the voltage *levels* to MIL−188 specifications.

5.11 Sample Line Setups

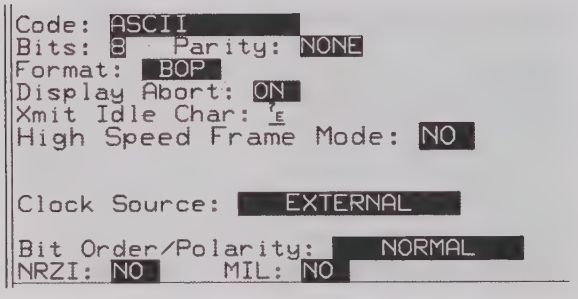
Figure 5-11 shows typical Line Setup configurations for various protocols.



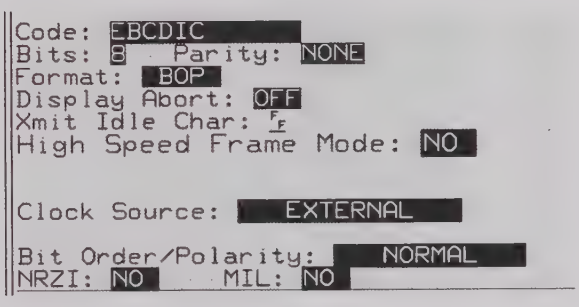
ASYNC



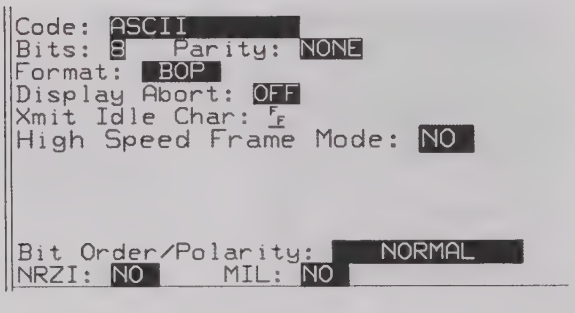
BISYNC



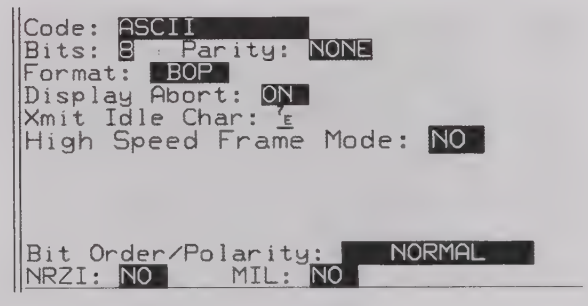
X.25



SNA



ISDN



SS#7

Figure 5-11 Sample Line Setup configurations for the following protocols: *X.25, SNA, ASYNC, BISYNC, ISDN,* and *SS#7.*

6 Run–Mode Display

Display Setup

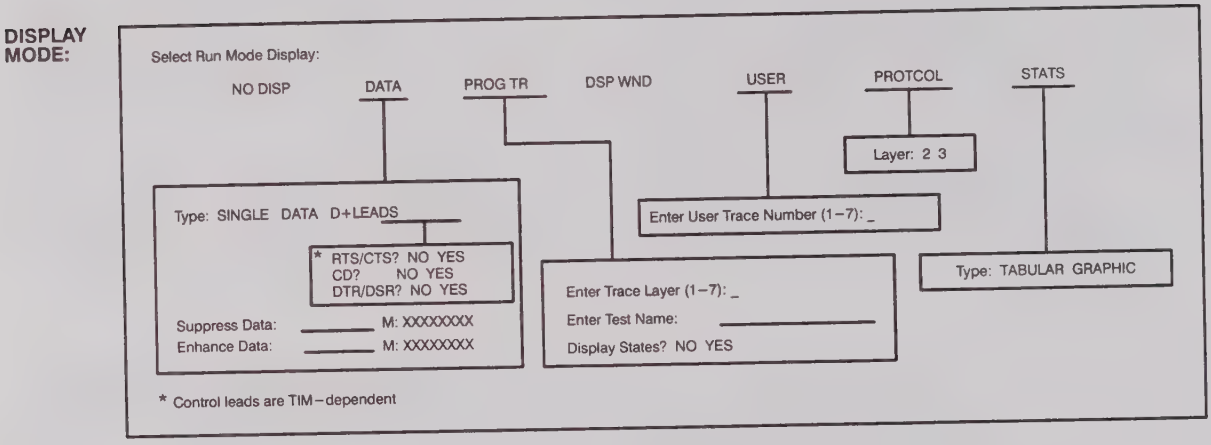


Figure 6-1 The Display Setup menu.

6 Run–Mode Display

Press [RUN] to compile and execute a test program. The compilation phases are discussed in Section 2.6.

If AUTOMON is the **Mode** selected on the Line Setup menu, the compilation will be delayed until the unit has configured its line setup parameters automatically.

6.1 Auto Configure

When the operator presses [RUN] with **Mode:** AUTOMON (the default mode) selected on the Line Setup menu, an Auto Configure program is loaded in that displays the screen in Figure 6-2. A Status field appears at the bottom of the screen, identifying the stage in the auto–configure procedure. The first stage is a clock–sampling phase that is reflected in the Status field as Capturing Sample #1. Once clock source and clock speed have been determined, these parameters are posted to the screen (see Figure 6-2) and the autolearn process moves to its second phase, Capturing Sample #2.

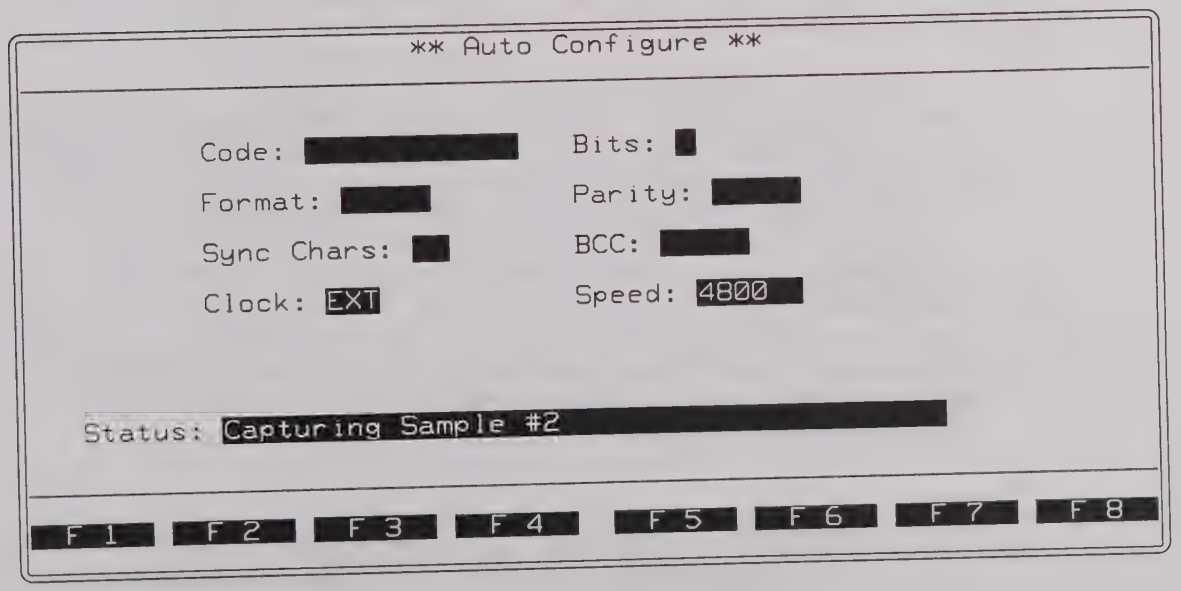


Figure 6-2 *Clock* and *Speed* are analyzed during the capture phase.

The unit remains in this stage from 1 to 6 seconds, depending on the data speed—one second for data captured at 19.2 Kbits per second or higher, a longer duration for captures at lower speeds. When the sample period ends, the sample is evaluated for its suitability as the basis for a new configuration.

If the sample is adequate—if it contains a high enough number of "meaningful transitions" and meets other similar criteria—the unit enters its configuring phase and arrives quickly at an estimate of all the line setup parameters.

The unit does not try to configure itself if the first sample does not meet the criteria for adequacy. Rather it returns to sampling data, this time Capturing Sample #3; and so on until a good data sample is found.

> **NOTE:** If an autoconfiguration is attempted on a line with no data or clock, the data–sampling will go on indefinitely. If the operator presses PROGRAM and checks the Line Setup menu, the **Clock Source** field will be set at INTERNAL and the **Speed** at 168000 bits per second. This is the first (highest) internal clock speed that is tested. Without data, the software will have had no reason to reject this speed and try the next lower one.

The initial data sample is very important. If at any time you wish to restart the autolearn process and try for a "better" initial data sample, press PROGRAM to terminate Automonitor and press RUN to begin again.

Clock and **Speed** (see Figure 6-2) are analyzed during the capture phase. Other parameters are analyzed during the configuring or "learning" phase. When the unit has arrived at an estimate of all the parameters, the Status changes to Unit Configured. The complete set of parameters is displayed for one second (see Figure 6-3), and the selections on the Line Setup menu screen are updated. After one second, the unit shifts to Run mode, compiles the program that is entered on the setup screens (including the new Line Setup), the Trigger Menus, and the Protocol Spreadsheet, and begins monitoring with its new configuration.

If the operator presses PROGRAM while the unit is in the learning phase, the Line Setup fields will be updated to include any parameters already registered on the Auto Configure screen. Pressing RUN, however, will always initialize the autolearn process.

> **NOTE:** In order for the unit to configure itself successfully to monitor synchronous data, the sync characters should be preceded by two or more idle–time characters.

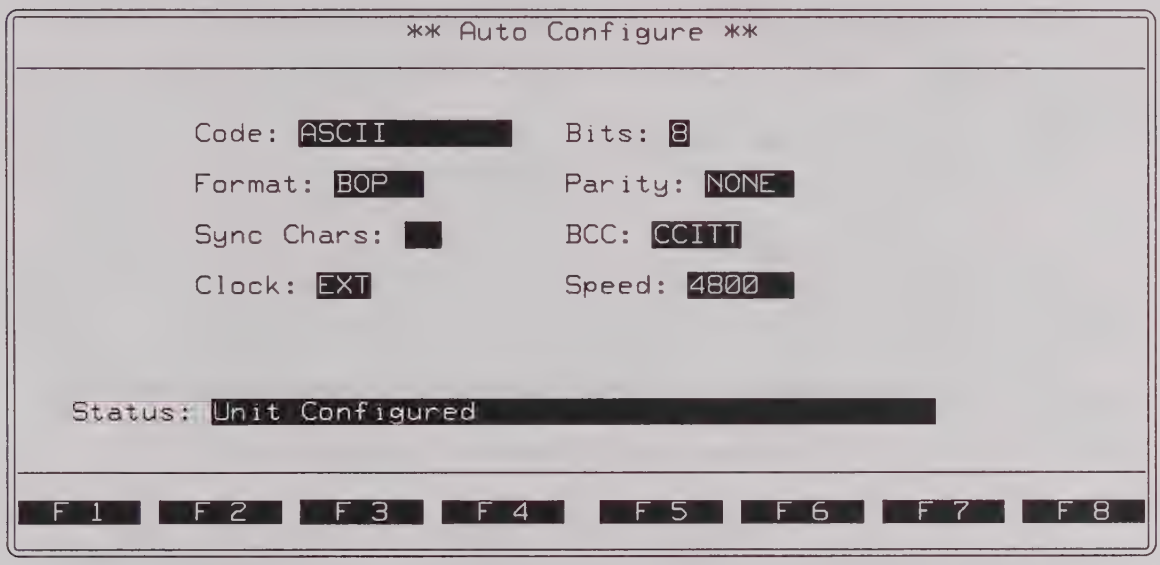


Figure 6-3 The entire set of parameters is displayed when autoconfiguration is completed.

The INTERVIEW's Automonitor mode has been designed to be (1) fast, and (2) 90% accurate. To satisfy the first criterion, the software acts on two assumptions: first, that the protocol is a fairly typical one; and also that the data sample is typical. (For example, an X.25 line without traffic cannot be distinguished from an SDLC line that also merely is idling 7E flags. In the absence of valid data, the Automonitor logic will not configure the unit.) Automonitor is designed as a tool for setting up a piece of test equipment properly, to be used with other tools. These tools include the Line Setup menu and the operator's own knowledge and experience.

6.2 Entering Run Mode

Figure 6-4 shows the screen of the INTERVIEW after Run mode has been entered. In any display mode, the *data* screen has three basic divisions.

The top two lines of the screen are status lines that provide setup information including test mode (monitor, emulate, or BERT), data source (line or disk), disk drive (hard, floppy 1, floppy 2—when source is disk), block number (for data recording or playback), record/playback status ("R"ecording line data, "P"laying back disk data, or "S"uspended record/playback), date and time, code, number of data bits, parity, synchronization format (synchronous, BOP, or async/isoc), and sync pattern if the format is synchronous. (The second line of the status area is also the prompt line.)

The bottom three lines of the screen are given to softkey prompts and a group of softkey functions that control the actual mode of display as long as the unit is in real–time or frozen Run mode. The remaining sixteen lines display test data or user–generated messages in a mode selected by the user.

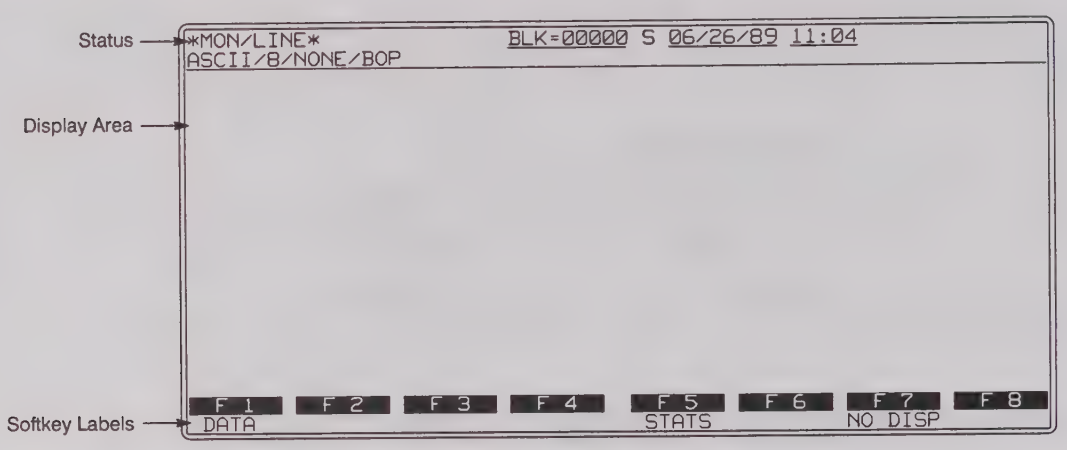


Figure 6-4 The three divisions of the data screen.

During programming, the user selects the display mode on the Display Setup menu (Figure 6-1), the top right sector of the Line Setup screen. In Run mode, the softkeys will change the display selection. Selectable display modes include character data, character data plus control-lead timing, protocol trace, program trace of state-to-state movement and of user-entered messages called "traces," and statistical tabulation or graphic display of counters and timers.

All data and statistics displays are available at *all* times during Run mode. Other special displays, Program Trace for example, are conditionally present. Figure 6-5 shows the two consecutive racks of softkeys that make the various data displays accessible to the user. To return to the first softkey rack, press [DONE].

> **NOTE:** An additional softkey, **QUAD**, is available for use with dual-port TIMs in the [F5] position using the *#pragma quad_display* directive in a spreadsheet program. This simultaneously displays TD and RD for both ports. For information on correct use of this directive, see Section 60.

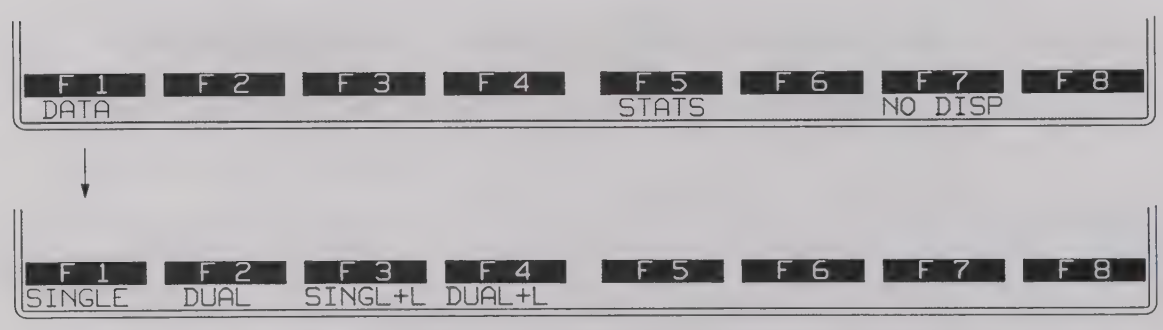


Figure 6-5 Two banks of softkeys give access to data display modes during a run.

There is also a **Display Mode** selection called NO DISPLAY. This suppresses the writing of data to the screen (though not to the screen buffer). See Section 6.11 for an explanation of this display mode.

The final **Display Mode** selection on the Display setup menu will be the first display mode presented to the screen in Run mode. All other available display modes can then be accessed in real time via Run–mode softkey.

Certain Display Setup menu selections cannot be changed in real time via softkey. Under **Display Mode:** DATA, regardless of the **Type** field selection, the **Suppress** and **Enhance** entries remain in effect during Run mode (except when operating in Frame mode, where these fields do not apply). When you choose **Type:** DATA+LEADS as the display mode, control–lead subfields appear. Only the control leads you select with a YES will be displayed during Run mode. Under **Display Mode:** PROG TRACE, you will see the subfield **Display States**. You may not change your selection for this field during Run mode.

6.3 Selecting Character–Data Display

Character data may be displayed in single or dual lines, with or without control–lead transitions. All four combinations of data leads and control leads are selectable on the lower rack of softkeys in Run mode. See Figure 6-5.

The **Type** of data display is also a selection field on the Display Setup menu. See Figure 6-6. The DATA softkey (Figure 6-5, top) enables the last **Type** option selected, even if that selection was dormant—not visible—on the Display Setup menu when the user pressed RUN. Other **Type** options can be accessed from the secondary rack of softkeys (Figure 6-5).

Fields on the Display Setup menu also accommodate lists of characters to be suppressed or enhanced.

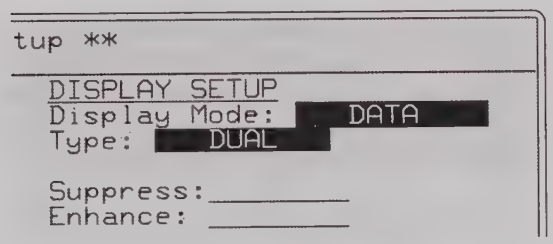


Figure 6-6 *Type* is a subfield under *Display Mode: DATA*.

NOTE: The **Type**, **Suppress**, and **Enhance** fields do not apply when you have chosen **High Speed Frame Mode:** YES on the Line Setup portion of the screen. Therefore, these fields will not be written to the Display Setup screen when operating in Frame mode.

(A) Single/Dual

In single–line display (Figure 6-7), DCE (RD) data and DTE (TD) data are displayed alternately on the same line. They are easily distinguishable because DCE data is always underlined. Single–line is the display mode that is most economical of screen–buffer space.

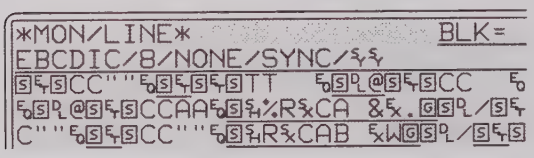


Figure 6-7 Single–line display.

In dual–line display (Figure 6-8), DTE (TD) data begins on the first data line. DCE (RD) data (always underlined) is shown on alternating lines beginning with the second data line. When one of the data leads is idle, time correlation with the other lead is maintained by this time–fill symbol: ⁖. See Figure 6-8.

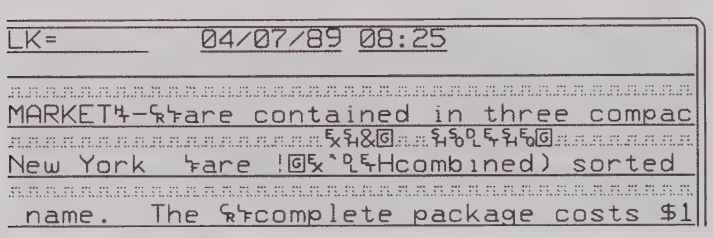


Figure 6-8 Dual–line display.

Select DUAL whenever the communications are full duplex. Single–line display is not appropriate when data is flowing in both directions simultaneously. The single–line display will preserve timing relationships to within one character time, with the result that the data will be less easy to read. See Figure 6-9, where an RD (underlined) message is interrupted by alternating TD characters.

In both single and dual display, fresh data is written to the screen beginning at the top left and moving to the right before retracing horizontally, and down to the bottom of the screen before retracing vertically. At a given moment during real–time display, there are always two blank lines on the screen. These two lines move just ahead of the freshest data and continuously overwrite the oldest data.

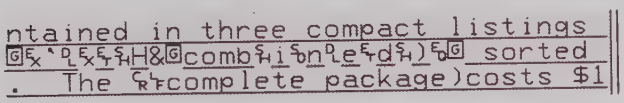


Figure 6-9 Avoid single–line display for full duplex data.

(B) Data Plus Leads

By pushing the softkey labeled SINGL+L or DUAL+L (Figure 6-5), the user can monitor not only two data leads, but also a timing pattern for up to five control leads. A display of single–line data plus all five leads is illustrated in Figure 6-10. The two states of the timing pattern can be defined in visual terms as low/high, in "handshake" terms as off/on, and in electrical terms as minus/plus voltage.

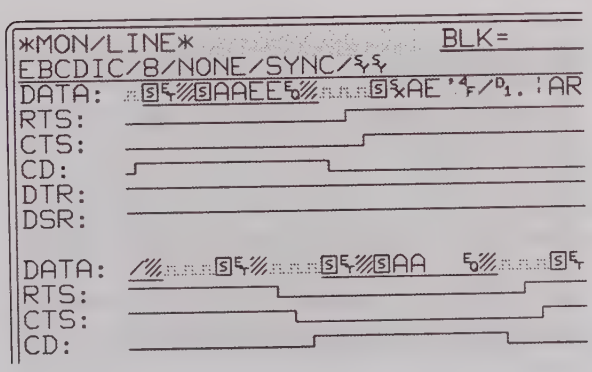


Figure 6-10 Control–lead transitions are displayable.

Control leads are selected for display on the Display Setup menu. EIA leads available for screen monitoring are RTS–CTS (selectable as a pair), DTR–DSR (also selectable only as a pair), and CD (carrier detect). See Figure 6-11 for the selection subfields under **Type:** DATA+LEADS .

If control leads are not set to be buffered on the Front–End Buffer Setup screen, control–lead status will not be available for data–plus–leads display and triggering. See Section 9.1(B) on buffering control leads.

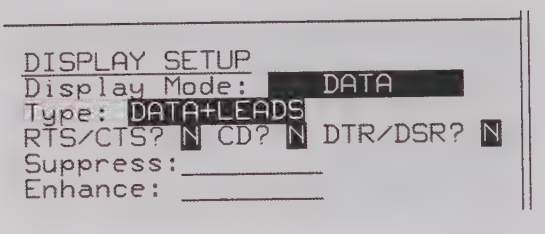


Figure 6-11 Five control leads are selectable under *Type: DATA+LEADS.*

A full set of leads is written to the bottom of the sixteen-line display area only if it fits completely. A full set is one or two lines of TD/RD data (with RD underlined) and a state-and-timing line for each of the control leads specified previously on the Display Setup menu. If a set is six lines and only four display lines remain on the screen, the entire set is displayed at the top of the next screen.

If the set is comprised of six lines (one data line and one line each for five control leads), two full sets of lines will be written to each successive screen. If a full set equals three lines, five sets will be written per screen.

The purpose of the data-plus-leads display is to show the *sequence* of events. Two data bytes (or a data byte and a control-lead transition) are never displayed in the same vertical column. Otherwise, the order of their occurrence would be lost to the display. Even if the events were detected a millisecond or a microsecond apart, they are displayed in sequence.

Turn **Display Idle:** ON if you wish to preserve a visual record of time intervals between lead transitions.

Precise timing intervals, to a resolution of ten microseconds, between lead and data and between two leads can be attained both for live and recorded data with a simple trigger program that uses timers or via cursor timing on the data-plus-leads display. See Section 6.4(E). Timer increments are discussed in Section 9, FEB Setup; and Section 21, Tabular Statistics.

(C) Suppress

A data-entry field labeled **Suppress** appears on the Display Setup menu along with **Display Mode:** DATA. For single- or dual-line display mode, you may choose up to eight characters to be suppressed from the screen display. The suppressed characters will not appear on the screen or be available in the character buffer for playback. They will, however, be considered by the triggers, included in counting and timing where applicable, and recorded if bit image data is being recorded.

The characters to be suppressed are entered directly from the keyboard and may include:

- Upper– and lower–case alpha characters and numerals;
- Control characters;
- Hexadecimal entries;
- Flag bytes (BOP format only);
- One bit mask;
- All characters not equal to a given character or bit mask.

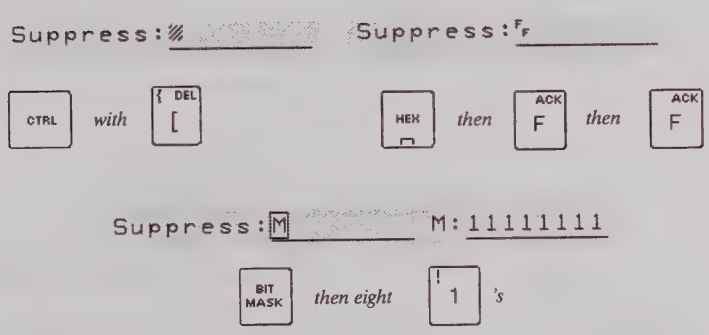


Figure 6-12 *FF* idle can be suppressed as a character, a hexadecimal, or a bit mask.

Figure 6-12 shows three different ways to suppress idle $^{F}_{F}$ on the data display. The idle character may be entered as an ASCII character, a hexadecimal, or a bit mask. Use [HEX] to turn on the hexadecimal function for all hex entries. Press [BIT MASK] to bring up a string of X's that you may overwrite with ones or zeroes.

Figure 6-13 shows the **Suppress** field when $^{7}_{E}$ flags are suppressed. Note that the [FLAG] key must be used for this entry. If literal $^{7}_{E}$ is entered with hex turned on, the logic will not read this as a flag, but instead as data (zero–stuffed) $^{7}_{E}$.

You can use the [NOT EQUAL] key–function in the **Suppress** field to *display only* a specific character (or set of characters represented by a bit mask). Press [NOT EQUAL] followed by one ASCII character, hex character, or bit mask. The suppressed character will appear in the **Suppress** list with a horizontal bar through it to indicate "not equal."

Figure 6-13 The *flag* key must be used to suppress *7E* flags.

Figure 6-14 gives two examples of the Suppress Not Equal ("display only") function. The top entry is used to display only the X–OFF (DC3) character. The Not Equal Bit Mask on the bottom will display four possible bytes only: $^{1}_{1}$ or $^{9}_{1}$ (DC1 in ASCII code) and $^{1}_{3}$ or $^{9}_{3}$ (DC3).

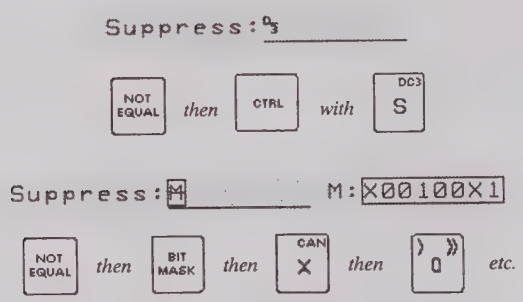


Figure 6-14 Display *only* X–OFF (top); display *only* X–ON/X–OFF (bottom).

When two or more Not Equal entries are combined in the **Suppress** field, only the listed not–equal characters are displayed. Refer to Section 25.3(I).

> **NOTE:** The **Suppress** field does not apply when you have choose **High Speed Frame Mode:** YES on the Line Setup portion of the screen. Therefore, this field will not be written to the Display Setup screen when operating in Frame mode.

(D) Enhance

A data–entry field labeled **Enhance** appears on the Display Setup menu when the **Display Mode** is DATA. The enhancement that results on the data display is blinking reverse video, dark lettering inside a light rectangle. Up to eight characters may be enhanced, including ASCII–keyboard characters, control characters, hexadecimals, and one bit mask.

Figure 6-15 shows a bit mask that enhances control characters in EBCDIC code. (Control characters are the only EBCDIC characters with zero–zero as the two high–order bits.) Beneath the bit mask is a data sample in which screen–formatting orders (always control characters) within the 3270–data stream have been enhanced.

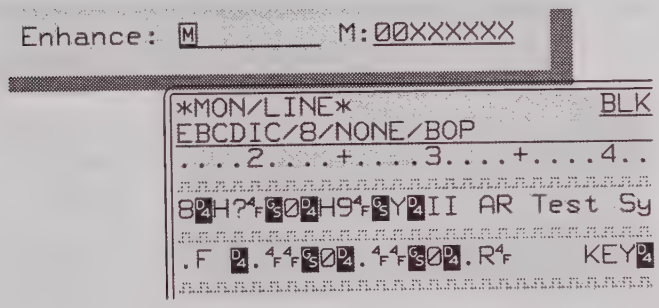


Figure 6-15 This bit mask enhances EBCDIC control characters only.

NOTE: The **Enhance** field does not apply when you have choose **High Speed Frame Mode:** YES on the Line Setup portion of the screen. Therefore, this field will not be written to the Display Setup screen when operating in Frame mode.

6.4 Special Features of Data Display

When character data is displayed on the INTERVIEW monitor, the bit stream is first divided into bytes according to the format (synchronous, bit–oriented, or start–stop) that the operator has selected on the Line Setup menu. Then the bytes are converted to displayable characters according to the code selected on the same setup menu. See Appendix D2 for the various byte–to–character translations. When a byte does not occur on the particular code–translation table and therefore cannot be decoded as a character, it is presented to the screen as a hexadecimal. All bytes are displayable, in hexadecimal at least.

In synchronous formats, display is suppressed automatically *before* synchronization and *after* outsync. Idle is usually suppressed, as a result of the receivers going out of sync; but idle display can be enabled on the Line Setup menu. In BOP (bit–oriented) formats, mark idle is suppressed automatically and cannot be displayed. $^{7}_{E}$ flags are not normally suppressed but they may be designated for suppression in the **Suppress** field on the Display Setup menu.

(A) Special Characters

Table 6-1 shows the special non–ASCII, non–Hex characters that appear on the data display. G and B are overlays that cover the last–received byte in the block–check (or frame–check) calculation. A is an overlay that *covers* the last full character

received before a BOP abort, defined as seven consecutive one–bits anywhere inside a frame. The [F] symbol represents a 7E flag, the literal (nonzero–stuffed) sequence 01111110 anywhere in the BOP bitstream. [S] stands for "sync." It indicates that the receivers have identified the sync pattern in the synchronous bitstream and locked on it as a new reference for character framing. The ... symbol maintains the time correlation between data leads when one of the data leads is idle.

Table 6-1
Special Display Symbols

Screen Symbol	Printed Symbol	Meaning
[S]	[S]	synchronization
[F]	[F]	BOP flag
[G]	[G]	good BCC or FCS
[B]	[B]	bad BCC or FCS
[A]	[A]	BOP abort
...	...	fill

(B) Hexadecimal Translation

In real–time or frozen Run mode, press [HEX] to display every data character on the screen as a hexadecimal. The hex translation is applied to the entire character buffer and not merely to characters written to the screen subsequent to the depression of the [HEX] key.

The hex display is an on/off function: press [HEX] again to restore normal decoding to the data. The LED on the cap of the [HEX] key lights up for hexadecimal data–entry, not display.

Hexadecimal notation divides each byte into two quartets or "nibbles." Figure 6-16 indicates the order in which the nibbles are transmitted/received when normal bit order is indicated on the Line Setup menu.

Note that the actual order of transmission of nibbles is not the order that we see when we read hexadecimal data casually from left to right: 2, 1, 4, 3, 6, 5, etc. Many protocol fields are designed around the *apparent* order of the nibbles rather than the actual order. The 12–bit logical channel number, for example, in an X.25 packet–header is composed of twelve bits that are not contiguous in the bitstream but appear to form contiguous nibbles in hexadecimal display.

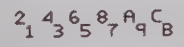


Figure 6-16 Hexadecimal series with digits indicating normal order of "nibble" transmission.

(C) Trigger Hex

Hexadecimal translation can be turned on and off by a trigger action on a Trigger Menu or on the Protocol Spreadsheet. Trigger hex is useful when you want to display some data in hex and some in translated characters.

In X.25, for example, frame and packet headers are bit–oriented and are more meaningful when they are displayed in hexadecimal. Once you are past the packet header in a normal (not "qualified") data packet, the data is character–oriented and the hexadecimal enhancement should be turned off. Figure 6-17 shows a display in which trigger hex has been used selectively on BOP data to convert protocol characters to hexadecimal. See Section 37.8 for the short trigger program that is controlling the display in Figure 6-17.

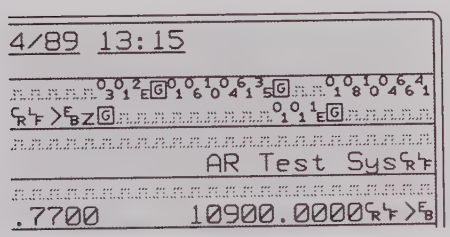


Figure 6-17 A set of four triggers is turning hex translation *on* for protocol characters and *off* for user data.

(D) Binary Expansion

By pressing FREEZE with character data on the screen, the operator freezes the display and activates the automatic character expander. Now the cursor appears on the monitor, under the control of the cursor–arrow keys. In the middle of the second line on the screen, the character at the cursor position is expanded in binary, with the DTE expansion on the left and the DCE expansion on the right. Figure 6-18 shows a simultaneous DTE and DCE expansion.

The rightmost bit in the binary expansion is the first bit received or transmitted when normal bit order has been selected on the Line Setup menu. The four right–hand bits compose the right–hand (lower) nibble in the hex byte that is expanded.

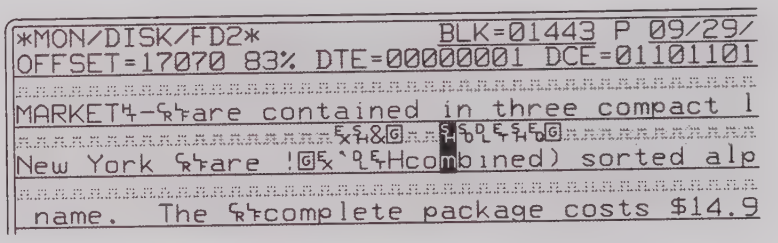


Figure 6-18 In Freeze mode, cursor characters are expanded in binary at the top of the screen.

(E) Cursor Timing

Cursor timing is another Freeze–mode display feature. It allows the operator to view the elapsed time between two events in the two data–plus–leads buffers and the single–line data display. It is not available in the dual–line data display.

To use cursor timing, *first enable time ticks* on the Front End Buffer Setup screen. Refer to Section 9.

The cursor–timing display is enabled/disabled via the MARK key. When you first enter Freeze mode, cursor timing is disabled. To set the location of the first event, press MARK. The character in the marked position is replaced by the M symbol. (In the data–plus–leads displays, M appears in all rows of the display.) Now use the cursor keys again to move the cursor to the next event. The elapsed time between the marked event and the cursor event is shown in the upper right–hand corner of the display screen. See Figure 6-19. The unit of time used in the cursor–timing display is the same as the one you selected when you enabled time ticks on the FEB Setup screen.

> **NOTE:** The sign the time value takes depends on which direction the cursor moves from the marked event. If the cursor moves forward from the marked event, the elapsed time is displayed as a positive value. If the cursor moves backward from the marked event, the time value is negative.

Each time you press MARK, you move the marked position—i.e., you redefine the first event. The previous M symbols are removed from the display and new ones appear at the cursor location. To disable the cursor timing display, return the cursor to the marked character and press MARK.

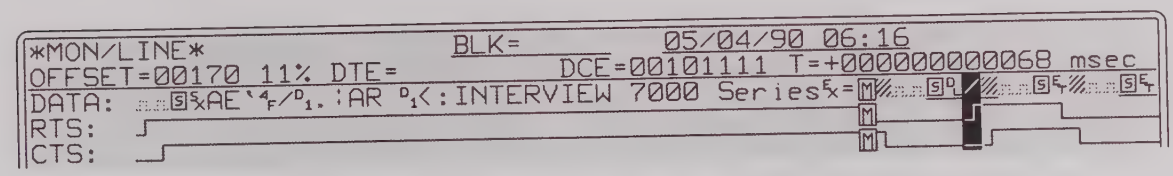


Figure 6-19 In this single–line, data–plus–leads display, cursor timing is used to display the elapsed time between state changes in the RTS lead.

6.5 Protocol Trace

The Display Setup menu and the softkeys associated with Run mode can present data to the user in forms other than raw character form. One of these alternate displays is a protocol trace. A protocol trace is enabled when a protocol "personality package" is loaded into the system from a disk via the Layer Setup screen. See Section 8.1, Personality Packages. Figure 6-20 shows sample data from a protocol analysis for X.25 Layer 3.

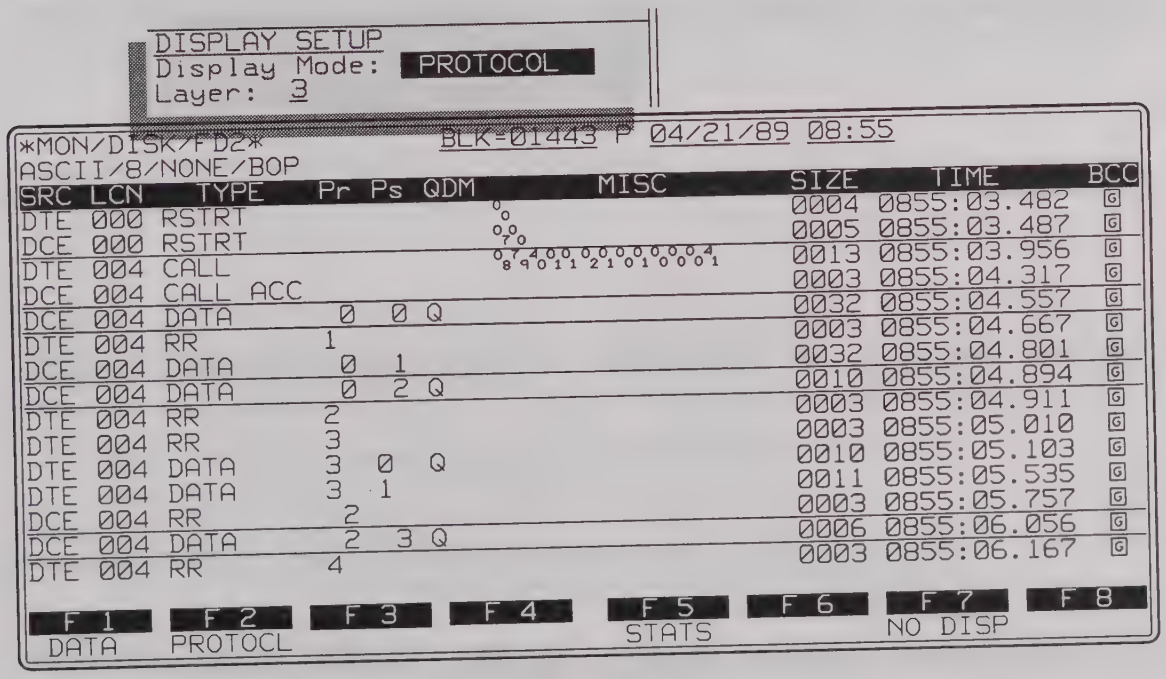


Figure 6-20 Protocol analysis for X.25 Layer 3.

To display the data in Figure 6-20, the operator first loaded in the X.25 Layer 3 package on the Layer Setup screen. Then **Display Mode:** PROTOCOL and **Layer:** 3 was selected on the Display Setup menu. When the INTERVIEW entered Run mode, the Layer 3 analysis was the active display mode.

If another display mode had been selected on the Display Setup menu, the Layer 3 analysis still would have been accessible after RUN was pressed. The PROTOCOL and **Layer** selections merely designate the display mode entering a run. They do not limit the display options during the run.

To access the Layer 3 protocol trace during Run mode, press PROTOCL on the first rack of softkeys. See Figure 6-21.

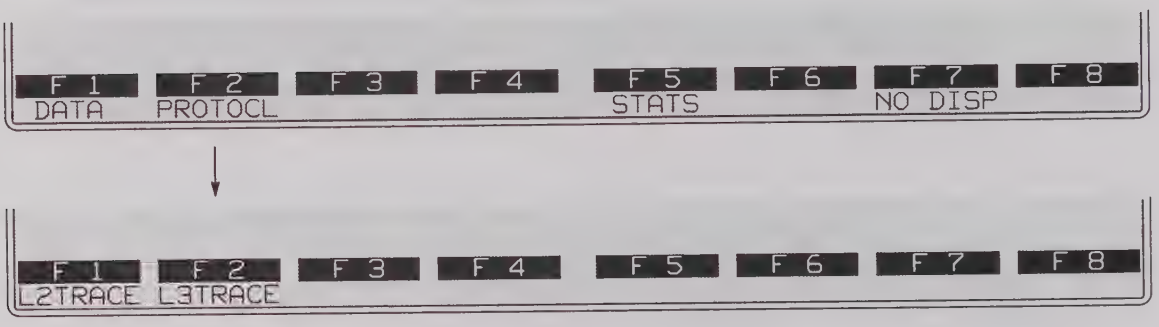


Figure 6-21 To access Layer 2 and Layer 3 protocol traces, press *PROTOCL*.

The location of the L2TRACE softkey depends on the layer packages loaded. When a Layer 2 package is loaded without a Layer 3 package, L2TRACE will appear on the first rack of softkeys. See Figure 6-22.

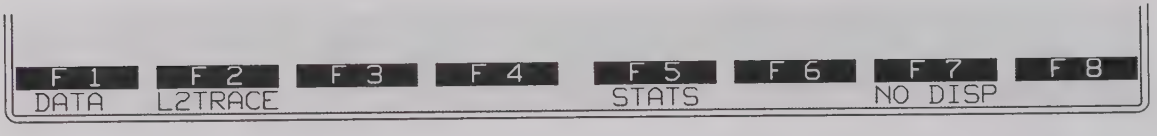


Figure 6-22 When a Layer 2 package is loaded without a Layer 3 package, *L2TRACE* will appear on the first rack of softkeys.

If both Layer 2 and Layer 3 packages are loaded, L2TRACE and L3TRACE appear on the second rack of Run−mode softkeys, as shown in Figure 6-21.

Note to C Programmers: You can modify Run−mode softkey labels for the Protocol Traces via the *set_ltrace_fkey_label* routine. See Section 60.7(C).

6.6 Program Trace

Program Trace is another mode of data presentation. See Figure 6-23. The Run–mode softkey PROG TR is present if you enter a TRACE spreadsheet action, if you write to the trace buffer *prog_trbuf* in C (see Section 60 for an explanation of trace routines in C), or if you select **Display States:** YES on the Display Setup menu.

As a program moves from state to state within the various tests and layers, the INTERVIEW maintains a log of user–assigned state and test names and user–entered messages called "traces." When the Program Trace is accessed (entering Run mode or via softkey) the sixteen data lines are dedicated to this log of states and traces. Unless its size is increased, the Program Trace buffer maintains a maximum of 4096 characters, equivalent to four full screens when every character space is used. (The size of the Program Trace buffer may be increased to a maximum of 16,381 elements via the *#pragma tracebuf* preprocessor directive, also discussed in Section 60.) When this limit is reached, new characters written to the end of the buffer force out the same number of characters from the beginning of the buffer. In Freeze mode, you may scroll through the Program Trace buffer via the ↑, ↓, ROLL FWD, ROLL BACK, PREV PAGE, NEXT PAGE, and HOME keys.

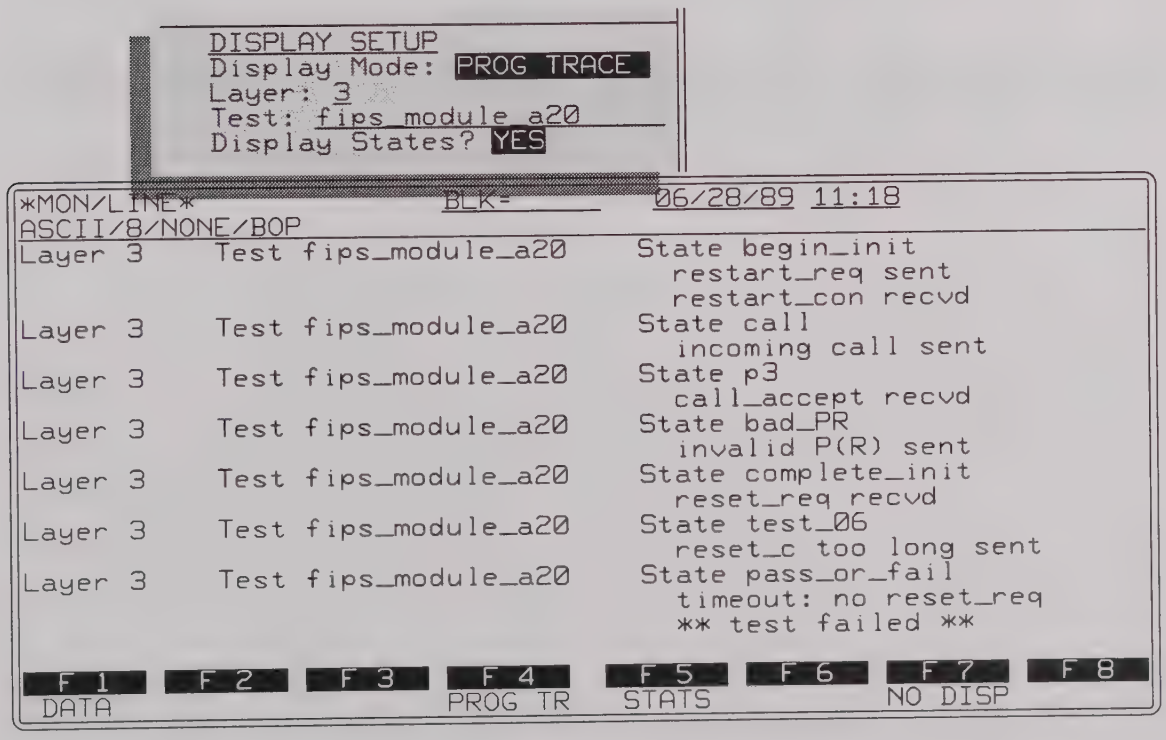


Figure 6-23 The user may select a specific layer and test for a Program Trace.

(A) Layer, Test, and State

States are the third level from the top in the programming hierarchy on the Protocol Spreadsheet screen. Inside states are *triggers*, comprised of *conditions* and *actions*. Above states are *tests*. At the top of the hierarchy are *layers*, corresponding to OSI protocol layers. See Section 24, The Layered Program Model.

Numerous layers and numerous tests per layer can be active concurrently in the INTERVIEW. The Program Trace can be set up to track state–to–state movement only in a particular **Layer** and **Test** identified by the operator on the Display Setup menu. Figure 6-23 gives the Display Setup necessary for the Program Trace shown. In the default setup, the **Layer** and **Test** fields are blank. In this configuration, the Program Trace will track all tests and layers.

Traces are trigger actions in Protocol Spreadsheet programs (see Section 31, Layer–Independent Conditions and Actions). They are simply user–entered ASCII data strings, identical to prompts except in their mode of display: traces are posted one to a line in the sixteen–line Program Trace display, while prompts appear on the second status line in all data–display modes (including the Program Trace). At a given moment during real–time display, there is always one blank line on the screen. This line moves just ahead of the freshest trace message and (once the trace buffer is full) continuously overwrites the oldest one.

State names can be included in the Program Trace via the **Display States:** YES selection on the the Display Setup menu (Figure 6-23). You will find the state log highly useful for locating dead ends, states that the test can enter but cannot exit due to a programming glitch. Traces are debugging tools, also. Inside a dead–end state they can inform you whether a particular condition that you are expecting is coming true.

Traces also allow you to keep a record of selected protocol events—to design your own protocol analysis. Figure 6-24 shows a user–designed trace for X.29 and X.3 protocols. Unlike prompts, traces are not immediately overwritten by other traces, so they are highly useful when you are trying to track protocol events that occur in quick succession. In Freeze mode, you may use the cursor keys to scroll through the Program Trace buffer.

```
*MON/LINE*                    BLK=_____    06/28/89 11:18
ASCII/8/NONE/BOP
DCE
SET
  PAD Recall
    L
  Data Forwarding
    R
  Idle Timer Delay
    No data forwarding on timeout
  Ancillary Device
    Data transfer (X-ON)
  Service Signals
    Other than prompt (std form)
  Action On Break
    Int
    Ind Brk
    Discard output

 F 1    F 2    F 3    F 4    F 5    F 6    F 7    F 8
 DATA                 PROG TR  STATS        NO DISP
```

Figure 6-24 Program Trace for X.29 PAD parameters.

(B) Line and Percentage

When you press FREEZE, the second status line at top of the screen changes. Two new fields, LINE and PERC, replace the code, parity and format indications. These fields provide information about the location of the cursor within the data written to the trace buffer. As you scroll through the buffer via the cursor keys, the values in both fields change. LINE indicates the line number of the current cursor position in the data. PERC reflects the percentage of the data in the trace buffer past which the cursor is located.

6.7 Statistics Display

There are two statistics displays in the INTERVIEW, tabular and graphic. Both of these displays can be accessed by softkey at any time during Run mode. Figure 6-25 shows the softkeys for TABULAR and GRAPHIC statistics displays.

Figure 6-25 The two types of statistics display are accessible in Run mode on the second rack of softkeys.

When statistics are displayed in tabular form, horizontal rows in the table are labeled with user–assigned names. Each name represents a counter, a timer, or a set of counters or timers combined for statistical purposes in an "accumulator." The first column next to the name contains the current *value* of a counter or timer. (Accumulators neither count nor time and therefore have no current value.) The values in the next four tabular columns (Last, Minimum, Maximum, Average) are derived from *previous* current values and are updated each time the counter or timer is sampled (read and reset). The tabular display is illustrated in Figure 6-26.

MON/LINE BLK= 06/28/89 11:18
ASCII/8/NONE/BOP

Name	Current	Last	Minimum	Maximum	Average	Unit
hostfrm	106	16	2	256	84.03	
goodfrm		16	2	256	85.22	
badfrm		11	11	11	11.00	
abort						

Figure 6-26 The tabular display.

In the graphic display, the values that are shown as numbers on the tabular display—up to 48 of them, sixteen on the screen at any given time, selected by the user out of 400 possible values on the scrolling tabular screen—are represented as horizontal bars. See Figure 6-27. The two statistics screens are discussed in detail in Section 21, Tabular Statistics, and Section 22, Graphic Statistics.

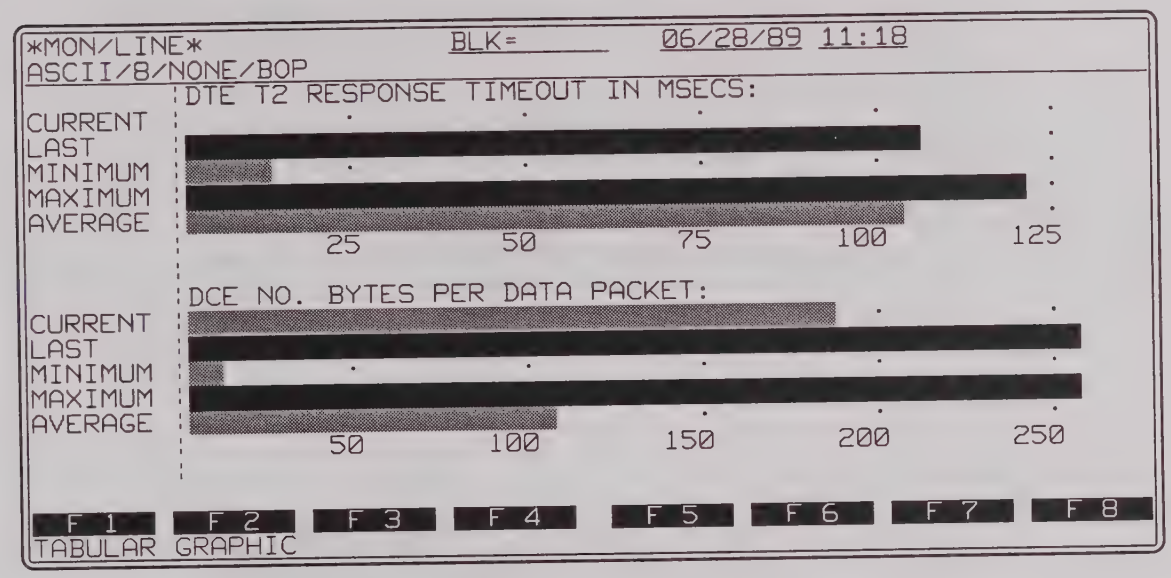


Figure 6-27 The graphics display.

6.8 Display Window

Figure 6-28 shows the Display Setup menu when **Display Mode:** DSP WND is selected. Display Window displays and preserves one screen, including the prompt line, of user–entered messages. When the end of the display screen is reached, the previous messages are overwritten, beginning at row one (the line below the prompt line). Messages are presented to the Display Window primarily via C display routines. See Section 60 for an explanation of these routines.

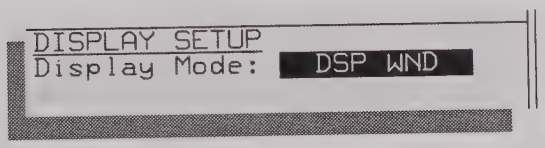


Figure 6-28 Display Window is a selection under *Display Mode*.

The Display Window lends itself to table or menu creation. The cursor may be positioned anywhere in the sixteen–line display area of the screen, or on the prompt line. Entries to a table, for example, may be updated by repositioning the cursor to a certain location.

Also, the programmer controls the use of certain Run–mode keys. The function of the cursor keys ↓ and ↑ can be programmed via the keyboard variables and the *send_key* routine explained in Section 68, Other Library Tools. (For other Run–mode screens, these keys control the playback speed of disk data.) For example, use ↓ and ↑ to move from field to field on a menu created in the Display Window.

Use the *set_dw_fkey_label* routine to assign a rack of Run−mode softkey labels to the Display Window. Additional C routines control the display and highlighting of these labels. See Section 60.4(C).

Figure 6-29 shows a Display Window created by the SNA Statistics application program (OPT−951−19−1).

```
*MON/DISK/FD1*          BLK=00100 P 08/23/88 10:04
==> active pu:C1    pu status:HOST_SENDING
                    SNA STATISTICS
                    Special Events
                    --------------

                        Primary     Secondary
      CRC errors              0             3
          Aborts              0             0
 Retransmissions             96             0
Negative Responses            7             0
Invalid FID format            0             0
-------------------------------------------------------------
   link startups:             1   reset count:          1
    elapsed time:           348 secs sample count:      77
last sense data:          0810000  from device:         %
        last FID:             2
        enter <S> for stats menu  <M> for main menu

 F 1    F 2    F 3    F 4    F 5    F 6    F 7    F 8
DATA                     STATS  DSP WND NO DISP
```

Figure 6-29 Display Window is useful for tables, as in this SNA Statistics example.

The DSP WND token, when present, is located on the first rack of Run−mode softkeys, as shown in Figure 6-30.

NOTE: In the absence of display routines (or softkey prompts) in a spreadsheet program, the Run−mode DSP WND token will not appear on any softkey rack. In this instance, if you select DSP WND as the display mode on the Display Setup menu, the Display Window will be the initial screen during Run mode, but it will be blank. If you move to a different display screen, you will not be able to return to the Display Window.

```
 F 1    F 2    F 3    F 4    F 5    F 6    F 7    F 8
DATA                     STATS  DSP WND NO DISP
```

Figure 6-30 Including display routines or *PROMPT* actions in a program causes the *DSP WND* token to appear in Run mode.

6.9 User Trace

There are seven trace buffers in addition to Program Trace. Select any one by specifying a user–trace number under USER TRACE display mode on the Display Setup menu, as shown in Figure 6-31. These buffers are similar to the Program Trace buffer. Messages are appended to the end of the buffer. Unless its size is increased, a user–trace buffer maintains a maximum of 4,096 characters, equivalent to four full screens when every character space is used. (The size of user–trace buffers may be increased to a maximum of 16,381 elements via the *#pragma tracebuf* preprocessor directive.) In Freeze mode you may scroll through the buffer using the cursor keys. The difference between user traces and Program Trace is that user traces are created only via C *tracef*, *tracec*, and *traces* routines. See Section 60 for an explanation of the trace routines and the *#pragma tracebuf* directive.

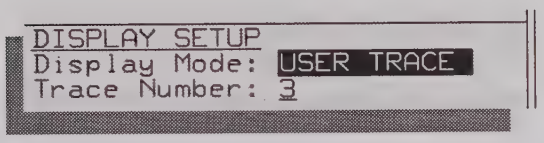


Figure 6-31 The initial display in Run mode will be user trace number three.

NOTE: In the absence of spreadsheet–program uses of user traces, a Run–mode trace token will not appear on any softkey rack. In this instance, if you select USER TRACE as the display mode on the Display Setup menu, the specified user trace will be the initial screen during Run mode, but it will be blank. If you move to a different display screen, you will not be able to return to any user trace.

When a user–trace buffer is written to in a spreadsheet program, a Run–mode token will appear for that buffer. The location of the token depends on the number of user buffers used. If only one user trace is used, a token indicating the trace number will appear on the first rack of softkeys, as shown in Figure 6-32.

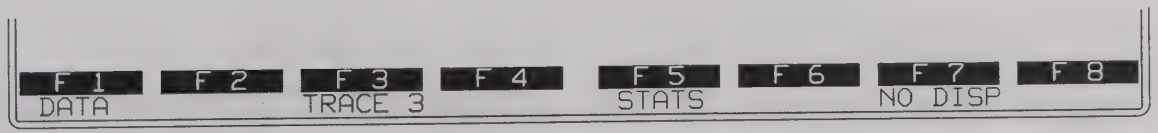


Figure 6-32 When only one user trace is used, its token appears on the first rack of softkeys.

If more than one user trace is written to, a USER TR token on the first rack provides access to the next rack containing tokens for all used buffers. See Figure 6-33.

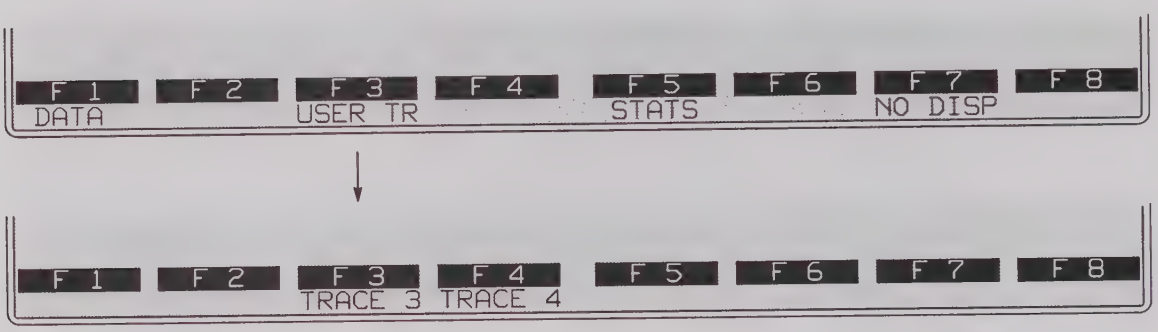


Figure 6-33 To access trace buffers when more than one has been written to, press *USER TR*.

Use the *set_utrace_fkey_label* routine to modify Run–mode softkey labels for the user traces. See Section 60.5(C).

Figure 6-34 shows a trace created via C *tracef* routines. Note the fields in reverse video, one of the attributes available to the user with *tracef* and *displayf*. Attributes are not available with softkey TRACE and PROMPT actions. Another advantage of the user (*tracef*) trace is that it can use the screen more economically. Where the softkey TRACE action assigns each message to a new line, *tracef* does not automatically generate a new line. A series of *tracef* messages may be written across the width of the screen. More information can be stored on a single screen.

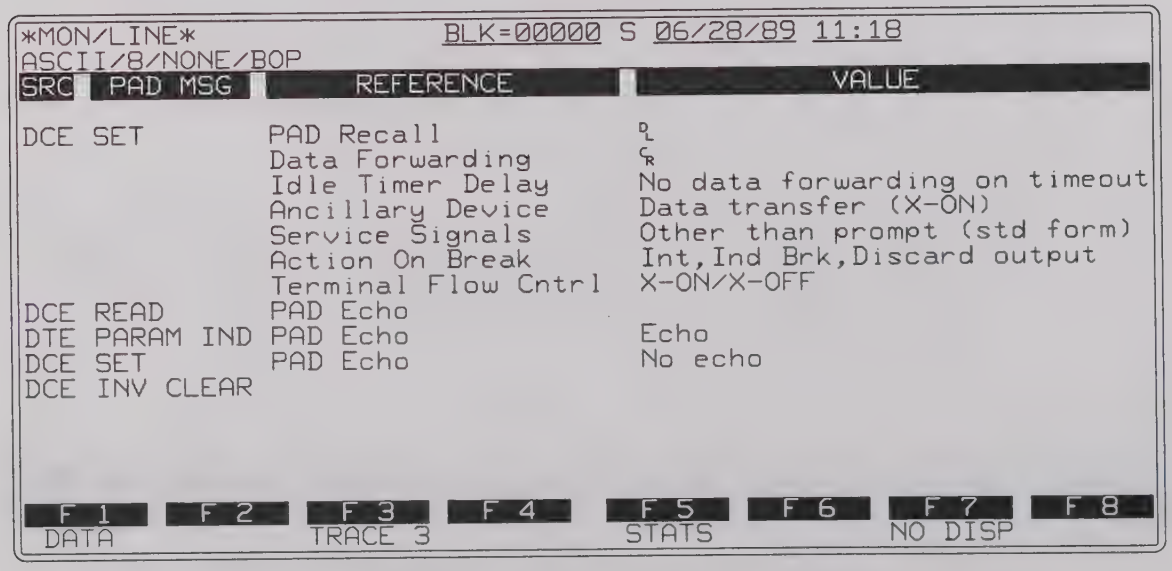


Figure 6-34 X.29/X.3 trace generated via C *tracef* routines.

NOTE: The heading displayed in reverse image in Figure 6-34 is written to the first line of the trace buffer. It is not a constant screen heading like those in the Protocol Traces. When you scroll forward through the buffer, the heading will disappear from the screen. It will reappear when you return to the beginning of the trace buffer.

6.10 Display Correlation in Freeze Mode

During Run mode, buffers are maintained for character data, protocol traces at each layer, one program trace, and seven user traces. When the operator presses FREEZE to stop the display, all of the buffers used are accessible by softkey and may be scrolled through. Certain buffers are correlated. The Layer 3 Protocol Trace buffer correlates to the Layer 2 Protocol Trace buffer. Either Protocol Trace buffer correlates to the 64–Kbyte character–data buffer. "Correlate" means that the packet displayed at the top of the Layer 3 Protocol Trace is contained in the frame displayed at the top of the Layer 2 Protocol Trace. This frame will begin—i.e., will be positioned in the upper left corner of—the corresponding data–display.

Figure 6-35 shows three Freeze–mode displays that relate to the same sample of raw data. The top display is a Layer 3 X.25 trace, the middle is a Layer 2 X.25 trace, and the bottom display is dual–line data. The operator uses the cursor–arrow, roll, and page keys to move freely around the buffers in Freeze mode. For example, when a softkey is pressed, to change the display from L2TRACE to DATA, the INFO frame at the *top line* of the Layer 2 Protocol Trace is also the first frame on the data–display screen.

(A) Offset and Percentage

OFFSET= and percentage readings are given on the second line of data displays in Freeze mode. Character–offset is the number of characters *previous to* the cursor–character in the 64–Kbyte character buffer. The percentage value immediately following the offset tells the percentage of the current buffer that contains data that was previous to the cursor character.

At two bytes of code and attributes per character, the character buffer holds an optimum number of 32,768 (or 32 K) characters. This translates to 32 screens of data in single–line display. Note, however, that if the Front–End Buffer Setup menu is configured to buffer time ticks, timestamps, or EIA leads (Section 9), the maximum number of characters in the character buffer will be reduced.

When the character buffer is full, it wraps to the beginning and new characters overwrite the oldest characters. The character buffer is likely to wrap much sooner than any of the trace buffers. There may be instances when a protocol element (a frame, for example) is shown on the trace display but the data version of the same element has been overwritten in the character buffer.

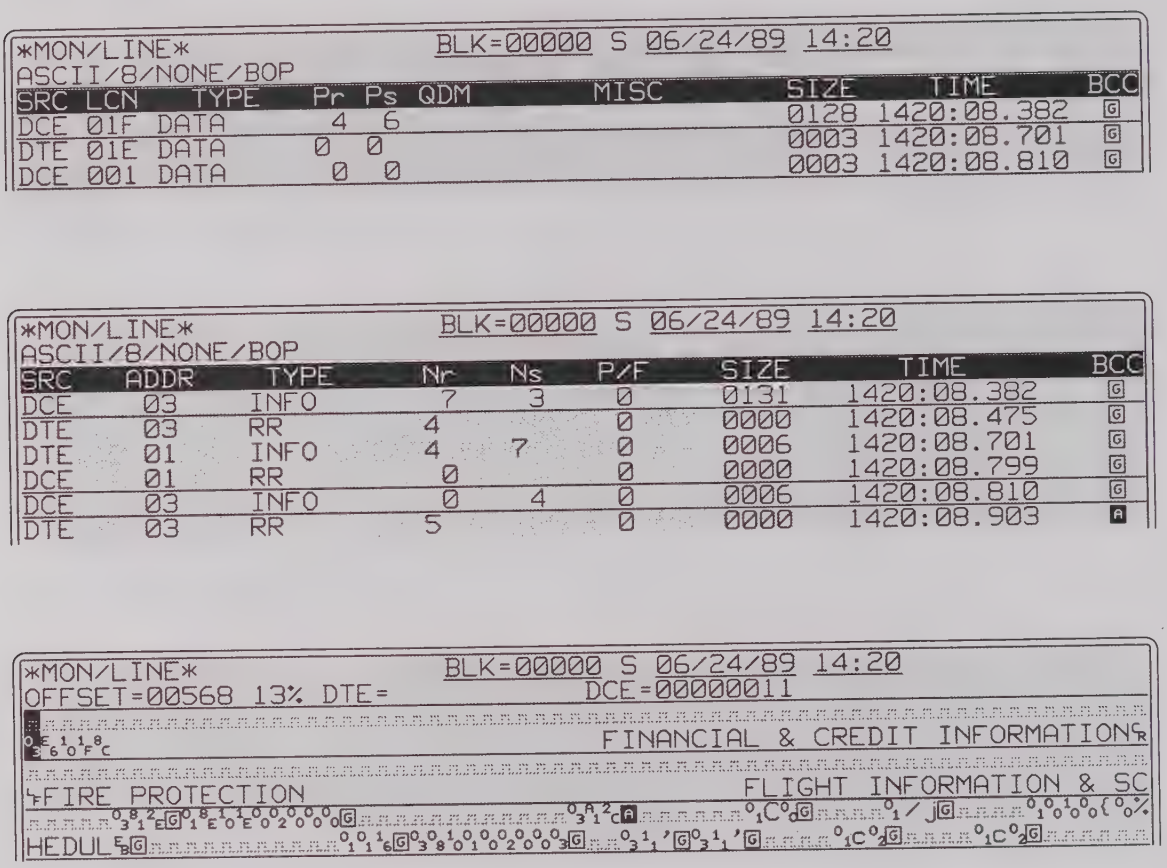


Figure 6-35 The first packet on the Layer 3 screen (top) correlates with the first frame on the Layer 2 screen (middle) and also with the first character–data frame (bottom).

6.11 No Display

If **Display Mode:** NO DISPLAY is the selection on the Display Setup menu, a blank screen will be displayed when the unit enters Run mode. The screen will remain blank until one of the display–mode softkeys (DATA, STATS, and so on) is depressed. Unlike Freeze mode, this mode does not prevent the writing of data to the screen buffer (also referred to in this manual as the "character buffer"). If you enter Run mode in No–display mode and then press a data softkey, all data will be present in the buffer (unless it is old data that has been overwritten).

The NO DISPLAY selection can be used to maximize efficient capture of data at very high speeds.

7 Record Setup

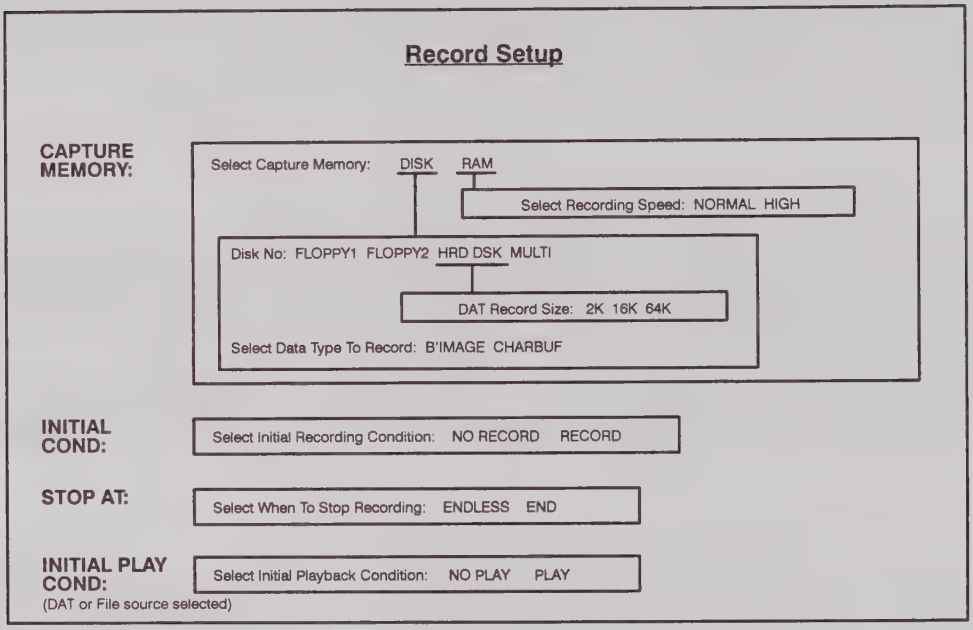


Figure 7-1 Record Setup menu.

7 Record Setup

The INTERVIEW provides storage of line data for later analysis. Recording of data can be initiated in three ways: through selections on the Record Setup menu, through manual control of the unit in Run mode, and through the RECORD action on the Protocol Spreadsheet. In addition, data is continuously saved to and can be retrieved from a temporary display buffer.

7.1 Format of Recorded Data

Data may be recorded to disk in one of two formats: as Character–oriented data, or as Bit–image data. It may be *recorded* to RAM as bit–image data only. Data is captured to character RAM (the screen buffer) automatically: this "capture" has nothing to do with the Record Setup.

(A) Character–oriented Data

Character–oriented data is data which is already formatted as specified by the user. This data is recorded as it was presented to the screen. (See Figure 2-7.) Character–oriented data is stored with programmed enhancements which may later be reviewed or replaced by the user. Superfluous data may be suppressed from character–oriented data, which allows more efficient storage. Except for code selection, line–setup parameters (described in Section 5) will not be applied to recorded character–oriented data.

(B) Bit–image Data

Bit–image data is bit–accurate data which is not preformatted. All line–setup parameters can be applied and reapplied to such data. This fact makes possible reinterpretation of data which would otherwise be unusable, because, for example, the wrong code, format, or block check calculation had been selected.

Bit–image data is stored prior to presentation to the screen. (Refer again to Figure 2-7.) This means that the data contains no enhancements and that it remains a complete record of all data as originally captured with the exception of idle, which can be discarded from the Front End Buffer. (See Section 9 for information on idle–suppression in the FEB.)

7.2 Recording Medium

Line data may be recorded directly to disk in bit–image format (and then may be transferred to a file), recorded to bit–image RAM and then transferred to disk or file, or captured in the screen buffer (character RAM) and transferred to disk.

The recording medium to be used in Run mode is chosen on the Record Setup menu. By default, data is not recorded when you enter Run mode. To activate recording from Run mode, press RECORD. (Otherwise, you may reconfigure the Record Setup menu to enter Run mode recording, or add a RECORD action to a Protocol Spreadsheet program.) At that point, data is recorded in the format and to the medium specified on the Record Setup screen. By default, data is recorded in character–oriented format onto the data acquisition tracks of FD1 (since MULTI is the default **Disk No** selection). If **Capture Memory:** RAM is selected, data will be recorded in bit–image format.

Before it is played back in Run mode, data stored in bit–image RAM must be transferred to disk or file. Go to the Disk Maintenance menu: press PROGRAM, F7, F3 (program, utilities, disk maintenance). Then execute a data transfer *from* RAM *to* disk or file. To transfer all or part of the RAM buffer, select **From:** RAM and **Type:** BIT IMAGE. Once the data has been transferred to disk or a file, monitor the data by selecting **Mode:** MONITOR and **Source:** DISK or FILE. See Section 5 for a further description of **Source** and other Line Setup selections.

> **NOTE**: Before data can be played back, all transmit actions must be cleared from the trigger menus, and all SEND actions and RECEIVE or RCV conditions must be deleted from the Protocol Spreadsheet.

7.3 The Screen Buffer

There is a third, alternative method of data capture always active in Run mode. A 64–Kbyte data buffer is provided for constant recording of character–oriented data as it is displayed on the screen. This means that, even though the operator has opted not to record data in real–time, the contents of the screen buffer can still be stored to a file, once the INTERVIEW has been placed in Program mode, and the data can be reviewed later. To control the capture of data to the screen buffer, use the **Capture** (ON / OFF) selection on the trigger–actions menu (see Section 26.9) or the Layer 1 CAPTURE action on the Protocol Spreadsheet [Section 32.3(F)].

Before it is played back in Run mode, screen data must be transferred from character RAM to the data acquisition tracks of a disk or to a file. Go to the Disk Maintenance menu: press PROGRAM, F7, F3 (program, utilities, disk maintenance). Then execute a data transfer *from* RAM *to* disk or file. To transfer all or part of the screen buffer, select **From:** RAM and **Type:** CHARBUFFER. The contents of the screen buffer can then be played back by selecting **Mode:** MONITOR and **Source:** DISK or FILE.

> **NOTE**: The screen buffer is erased each time RUN is pressed, so the data must be transferred to disk before the INTERVIEW returns to Run mode.

Of course, if the unit stays in Run mode, you can freeze the screen and review data without any transfer procedure. Freeze mode analysis is described in Section 3.4.

If you need to record a large amount of data for analysis, it might be preferable to employ the Record Setup features of the INTERVIEW, since the screen buffer is limited in size, and its contents are automatically overwritten once it is full.

NOTE: Data in the screen buffer is stored exactly as displayed. Suppressed data is not stored in the buffer and, as a consequence, is not available for triggering. Programming techniques must take this into account.

NOTE: Enhancements *are* stored with the data. The data is presented with the enhancements in playback, unless active triggers contain potential **Enhance** actions. If the INTERVIEW detects **Enhance** actions when a test is compiled, then old enhancements stored with source data are stripped to allow new enhancements.

7.4 SCSI Drive Maximum Recording Times

Small Computer System Interface (SCSI) drives enhance the maximum recording times in the INTERVIEW, as shown in the examples in Table 7-1. Note that the times in this table reflect a "worse case" scenario with every bit captured and idle suppress *off*; suppressing idle characters will increase recording times. Actual recording capacity is reduced by the megabytes allocated to the file system partition and other overhead.

Table 7-1
SCSI Drive Recording

Capacity	Maximum DAT Recording Time at 1.544 Mbps	Maximum DAT Recording Time at 2.048 Mbps
270 MB*	3.75 minutes	2.75 minutes
1.2 GB*	22.5 minutes	16.5 minutes

* These hard disk drives are representative; your drive may be smaller or greater depending on drive availability when your INTERVIEW was manufactured.

7.5 The Record Setup Screen

The parameters for real–time data capture are selected on the Record Setup menu, which resides on the same screen as the Line Setup and Display Setup menus. All possible Record Setup selections on this menu are shown in Figure 7-1. The default menu is shown in Figure 7-2.

NOTE: Bit–image data is subject to idle–suppression in the Front–End Buffer *prior* to recording. See Section 9 for details.

Figure 7-2 Default Record Setup screen.

(A) Capture Memory

Either DISK or RAM can be selected in this field.

1. *Disk.* DISK is the default selection. When DISK is selected, two additional fields, **Disk No** and **Data To Record**, appear immediately below. Refer again to Figure 7-2. See Section 1.6(D) on maximum line rates for recording data to disk. Also consult Section 2.10 for information on optimizing recording of data to disk.

   NOTE: Data can only be saved to formatted disks. See Section 14 for instructions on formatting.

   (a) *Disk no.* Available selections in the **Disk No** field are FD1 (Floppy Disk 1); FD2 (Floppy Disk 2); HRD (Hard Disk); and MULTI (Multiple disks). The default selection is MULTI.

   MULTI causes recording to begin on the floppy disk inserted in Drive 1 and to continue on the disk in Drive 2 when the first disk is completely full. Once this second disk is full, recording reverts to Drive 1, and the process continues. You may manually replace disks in the nonactive drive to record data alternately on a new disk. If no disk is inserted in the next drive, recording continues from the beginning of the last–recorded disk.

   CAUTION: Do not remove a disk from an active drive. You can always identify the active drive by the glowing LED just to the right of the drive.

   When HRD is the selected **Disk No**, an additional field is written to the screen beneath the **Data to Record** field: **DAT Record Size** (see Figure 7-3). This field

is discussed later in this section. Recording proceeds continuously or stops when the disk is full in accordance with the **Stop At** selection, described later in this section.

Figure 7-3 Record Setup screen configured for capture to hard disk with large record size

When FD1 and FD2 is the selected **Disk No**, recording proceeds continuously or stops when the disk is full in accordance with the **Stop At** selection, described later in this section.

(b) *Data to record.* The two selections in this field are CHARACTER and BIT IMAGE data. This selection specifies the format of recorded data. The differences between these two data formats are explained at the beginning of this section. This field disappears when **Capture Memory:** RAM is selected, since the format will always be bit–image.

(c) *DAT Record Size.* This field only appears when **Disk No:** HRD is chosen. Choose one of the three sizes: 2K (the default), 16K, or 64K. When 2K is chosen, the standard record size of 4 sectors (2048 bytes) is utilized when recording to the hard disk.

With **DAT Record Size:** 16K, the record size is 32 sectors (16384 bytes). This choice requires fewer "time–of day" stamps, allowing you to record at higher baud rates than the standard record size.

With **DAT Record Size:** 64K, the record size is 128 sectors (65536 bytes). This size is for use with recording speeds greater than 768 Kbps and *only with SCSI hard disk drives*—it is required for 2.048 Mbps recording. This record size requires less overhead for the CPM processor, allowing you to record at higher baud rates than the standard record size; however, it also requires a very large buffer area supported by the 6 Mbyte CPM processor boards, which are installed with SCSI hard disk drives.

NOTE: Data recorded to disk utilizing the larger DAT record size will retain this format even if transferred to a file or copied to DAT on another disk. System software version 10.00 or higher is required to monitor playback of data recorded with 32–sector records; system software version 11.00 is required for 128–sector records.

2. *RAM.* RAM refers to the portion of the INTERVIEW's internal memory reserved for bit–image data recording directly from the data line.

   When RAM is selected, the **Disk No** and **Data to Record** fields disappear. A new field, **Record Speed,** may appear. See Figure 7-4. If this field does not appear, RAM size is 1 Mbyte in the 8600 *TURBO*, 8700 *TURBO*, and 8800 *TURBO* and 256 Kbytes in the 8100 *TURBO* and 8200 *TURBO*. See Table 7-2.

Figure 7-4 Record Setup menu configured for high–speed recording.

(a) *Record speed.* The **Record Speed** field is present only when
   1) **Capture Memory:** RAM is selected, and
   2) the Line Setup **Mode** is MONITOR.

   There are two **Record Speed** selections, NORMAL and HIGH SPEED. The default entry is NORMAL. Normal recording can support a speed up to 256 Kbps. When line speeds approach this threshold, you may need to record according to the guidelines set forth in Section 2.10. High–speed recording is required when line rates exceed 256 Kbps.

   For normal and high–speed recording, RAM size is determined by the number of MPMs in your unit. See Table 7-2 for RAM size based on various INTERVIEW configurations.

   When HIGH SPEED is selected, record RAM is automatically increased to 1 Mbyte in the 8600 *TURBO,* to 2 Mbytes in the 8700 *TURBO*, and to 3 Mbytes in the 8800 *TURBO*.

   Choose **Record Speed:** HIGH SPEED when the line rate is between 256 Kbps and 2.048 Mbps full duplex. Two interfaces which may require high–speed record capability are T1 and G.703. To record the aggregate T1 data stream, for example, you must select HIGH SPEED. Refer to the accompanying documentation for these optional interfaces for more information on T1 and G.703 aggregate record. (There is no aggregate record for ISDN.) Note that ISDN, T1, and G.703 lines cannot be tested with the INTERVIEW 8100 *TURBO* unit.

Table 7-2
RAM Size (bytes) for Normal and High–Speed Recording in the INTERVIEW 8000 *TURBO* Series Units

Unit	Number MPMs	Normal	High–Speed
8100 *TURBO*	1 MPM	256K	256K
8200 *TURBO*	1 MPM	256K	256K
8600 *TURBO*	2 MPMs	1M	1M
8700 *TURBO*	3 MPMs	1M	2M
8800 *TURBO*	4 MPMs	1M	3M

At high speeds, data cannot be processed quickly enough to be displayed or monitored accurately. The HIGH SPEED selection, therefore, automatically turns off the monitoring function by disabling the line receivers. The following message is displayed on the status line of the Run–mode screen: *"Monitoring is disabled for high speed ram recording."* The display process continues, even though the line receivers are not supplying data for display. Stop the display process by selecting **Display Mode:** NO DISPLAY on the Display Setup menu, or by pressing NO DISP during Run mode.

Without active line receivers, data cannot be displayed and program conditions (or triggers) based on incoming data or leads can never come true. Data analysis must be performed during slower playback of recorded data. With this consequence of high–speed record in mind, you may still opt to use it at speeds less than 256 Kbps.

NOTE: For data to be available for playback, it must reside on a disk. Transfer data captured in RAM to a disk via the Data Transfer command on the Disk Maintenance screen. See Section 14.4(C).

High–speed recording formats data differently than normal recording. When recording ends (or you exit Run mode via the PROGRAM key), the data is converted from high–speed format to normal–data format. The following message is displayed: *"Reformatting ram record buffers – Please wait ..."*

(B) Initial Condition

The two selections available in this field are RECORD and NOT RECORD. When RECORD is selected, data recording begins the "instant" you place the INTERVIEW in Run mode. When NOT RECORD is selected, no recording takes place until one of three events occurs: a new program with **Initial Condition:** RECORD is loaded, a spreadsheet trigger activates recording, or the operator initiates recording manually. Trigger and manual control of recording are described later in this section.

A record/playback field (on the top line in the status area) on Run–mode screens will indicate whether recording is in progress or has been suspended. If recording is ongoing, an "R" appears next to the block–number field. If your initial condition is NOT RECORD or you suspend recording, "S" will be displayed. The field will be blank when the end of RAM or the data–acquisition tracks is reached. The record/playback field also will be blank if the **Capture Memory** field indicates that you will record to disk, but no disk is present in the selected drive or data–acquisition tracks are not available on the disk.

> **NOTE:** A "P" in the record/playback field indicates that data playback is in progress. See Section 5.2.

(C) Stop At

This field defines the action which the INTERVIEW takes when bit–image RAM or disk is full. ENDLESS LOOP causes recording to continue from the beginning of disk or RAM, whichever is designated as **Capture Memory**. ENDLESS LOOP causes old data to be overwritten. As a result, only the most current data is accessible for playback.

Stop At: END indicates that no further data will be recorded once bit–image RAM or disk is full.

> **NOTE:** This selection does not influence recording when MULTI is selected as **Disk No.** MULTI causes data to be recorded continuously, without stopping.

(D) Playback Control

When playing back recorded data in DAT or file mode, the operator can set the initial play condition of the INTERVIEW to either PLAY (the default) or NO PLAY. If **Initial Play Cond:** PLAY is selected, the unit automatically begins playback of

recorded data when it enters Run mode. If **Initial Play Cond:** NO PLAY is selected, the unit comes up in "suspended" mode—the top status line of Run–mode screens will show an "S" displayed in the record/playback field—and will only begin playback when a running program, such as one of General Signal Networks' application programs, issues a Resume command or the user presses RECORD.

Figure 7-5 Record Setup menu configured for playback from disk. The Initial Play Cond field appears when the Line Setup source is configured for DAT or file mode.

7.6 Trigger Control of Capture

Capture to the screen's character buffer can be placed under trigger control. The trigger action **Capture:** OFF (or the Layer 1 CAPTURE OFF action on the Protocol Spreadsheet) stops the buffering of character data and freezes the screen display. **Capture:** ON (or CAPTURE ON) restarts data display and capture. Capture can be applied to one or both sides of the display (TD and/or RD).

When the character buffer has been frozen by trigger rather than by the FREEZE key, you will not be able to scroll through data on the screen. See Section 26.9 or 32.3(F) for a further discussion of capture.

7.7 Spreadsheet Control of Recording

Recording to RAM or disk also can be placed under trigger control. Each recording session (i.e., you press RUN to begin the session) begins at block 0 on the disk. The layer–independent action RECORD OFF on the Protocol Spreadsheet stops the recording of line data. RECORD ON restarts recording. See Section 31 for a discussion of RECORD.

CAUTION: Once you stop recording, you must press the PROGRAM key in order for the INTERVIEW to update the data on disk. If you remove a floppy diskette from its drive (or power the unit off after recording to the hard disk) without pressing PROGRAM, the disk will not contain the data from the recording session.

You can ascertain whether data is being recorded by looking at the status lines at the top of the display. Incrementing block numbers and an "R" in the record/playback field (next to the block number) indicate that data is being recorded.

7.8 Manual Control of Recording

The operator can manually initiate recording in Run mode using the RECORD key. RECORD is a toggle key which will alternately start or stop recording to RAM or disk. The selections on the Record Setup screen and trigger actions influence the function of RECORD. Each recording session (i.e., you press RUN to begin the session) begins at block 0 on the disk.

> *CAUTION: Once you stop recording, you must press the PROGRAM key in order for the INTERVIEW to update the data on disk. If you remove a floppy diskette from its drive (or power the unit off after recording to the hard disk) without pressing PROGRAM, the disk will not contain the data from the recording session.*

During recording, the top status line of Run–mode screens will show incrementing block numbers and an "R" displayed in the record/playback field.

8 Layer Setup

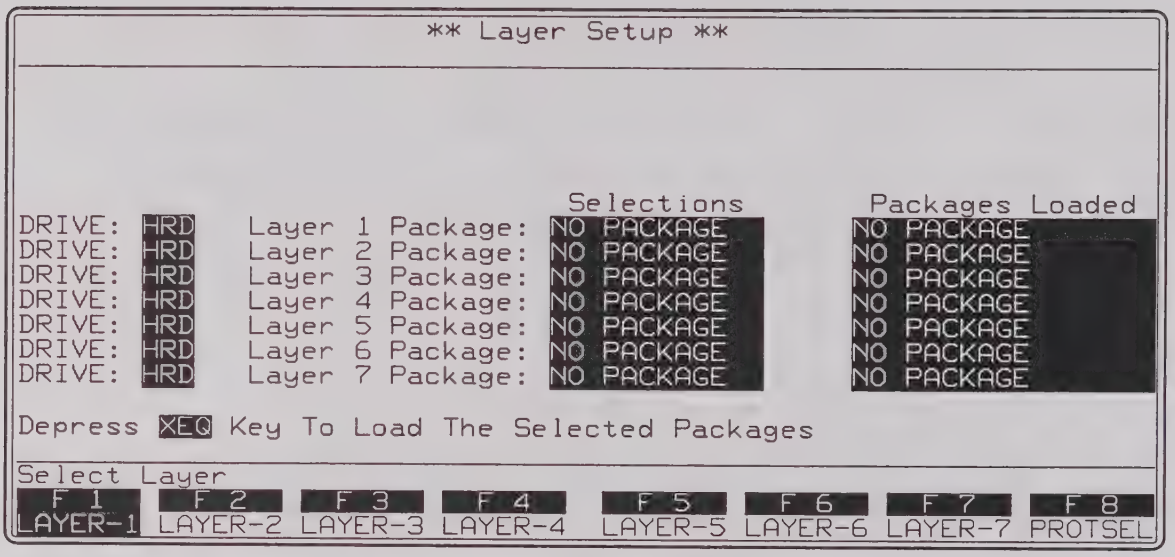


Figure 8-1 Default Layer Setup screen.

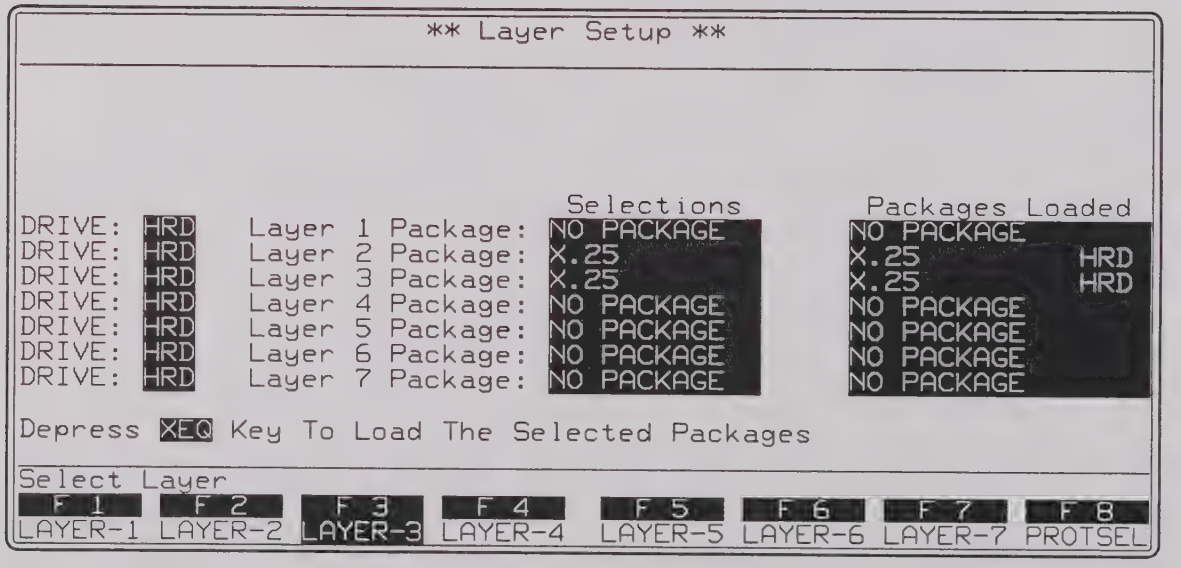


Figure 8-2 A Configured Layer Setup screen.

8 Layer Setup

The Layer Setup screen is directly accessible from the Main Program menu. A default Layer Setup screen is shown in Figure 8-1 and a configured Layer Setup screen in Figure 8-2.

> **NOTE:** Additional fields appear on this screen when the TIM expansion shelf (available as OPT−951−42−1) is installed. See the option's accompanying documentation for information on the menu fields and selections.

Most protocols available to the user also have certain definable parameters. These parameters are grouped on the Protocol Configuration screen described later in this section.

8.1 Personality Packages

The INTERVIEW provides layer−specific protocol packages called Personality Packages. These packages contain automatic selections for trigger conditions and actions. Automatic selections appear on the Protocol Spreadsheet once the Personality Package is loaded.

The Layer Setup screen gives the users access to the Personality Packages residing on disk in their system. Personality Packages are identified on the Layer Setup screen by protocol name and are categorized by layer and by the disk on which they reside. Since protocols are selectable by layer, it is possible to "mix and match" them.

A rotating field on the Layer Setup screen is assigned to each OSI layer. In the field, the user may choose from available protocols. The protocol chosen (displayed) in each field and then loaded will be the one used for monitoring and emulating that layer in Run mode.

8.2 Selecting and Loading Protocols

Two disks are supplied with the INTERVIEW or with software upgrades. The system disk (DSK−951−001−1.X) may be used for boot−up. The personality packages reside in the */usr/layer_pkgs* directory of the user disk (DSK−951−001−2.X). The same directory on the hard disk also contains the personality packages.

If your boot drive is one other than the hard drive (either Floppy Drive 1 or 2), place the user disk in the boot drive. Press CTRL – CLEAR to read the disk. Available personality packages will appear as softkey selections at appropriate layers.

For convenience, the OSI layers appear on the screen from lowest to highest. Access a layer in one of two ways: by moving the cursor up and down the Drive column with the ↑ and ↓ keys; or by pressing the function key for the appropriate layer when the prompt above the function keys reads "Select Layer." The **Drive** field for that layer blinks.

When you have selected a layer, press the appropriate function key or press ROLL BACK or ROLL FWD to display the options for the source drive. A rotating field at the left identifies the active drive for each layer. Once you have selected a drive (Hard Disk is selected for each layer in Figure 8-2), it is the only drive from which the Personality Packages for that layer can be loaded.

Only the names of the Personality Packages for the selected drive appear in the **Selections** column. Select the correct drive, then press DONE, RETURN, or → to move to **Selections**. Pressing the function key for a protocol or pressing the DONE key moves you to the rotating field for the next layer.

In the **Selections** column, display all the protocols you wish to use in Run mode. (If you don't want any protocol loaded at a certain layer, leave "NO PACKAGE" in its **Selections** field.) Then press the XEQ key. This loads all selected protocol packages from disk. Once the process is complete, the names of loaded protocols appear in the column labeled **Packages Loaded** on the right of the screen.

> **NOTE:** Protocols used should be compatible with the data format you have selected on the Line Setup screen (see Section 5). For example, if you are using SDLC, X.25, or SS#7 at Layer 2, you must have selected Bit–Oriented Protocol (BOP) as the Format.

8.3 The Protocol Configuration Screen

Typically, a protocol available on the setup screen has its own submenu, the Protocol Configuration screen, accessible from the Layer Setup screen. The contents of each protocol's configuration screen are explained in a section devoted to that protocol and layer.

(A) Accessing the Screen

Once the protocols you will be testing are loaded, you may call up the Protocol Configuration screen, in turn, for each protocol. Press PROTSEL. (You may always access the PROTSEL softkey by first pressing HOME.) Then select a layer by pressing the appropriate function key. This takes you immediately to the configuration screen for that protocol. If, for example, you are using X.25 at Layer 2, you will see the screen shown in Figure 8-3.

Make appropriate parameter selections. Return to the Layer Setup screen, if necessary, by pressing the PREV PAGE key. If you have additional parameters to set, press F8 (PROTSEL) and select another layer to call up the new parameters screen.

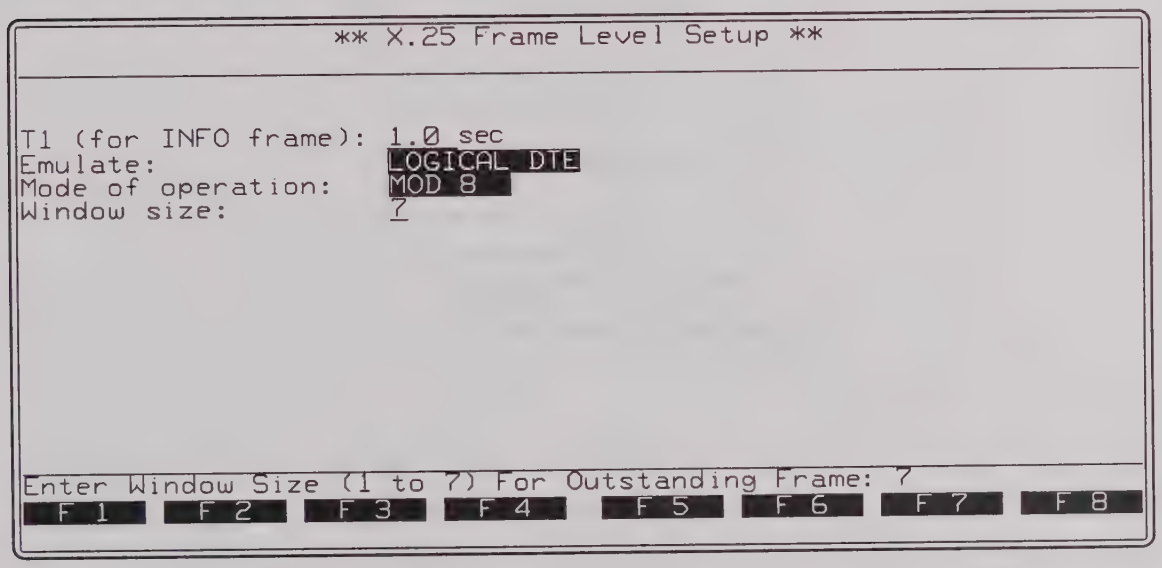


Figure 8-3 Most protocols have definable parameters, selected on a separate screen accessed from the Layer Setup screen.

(B) Default Parameters

Default parameters, loaded with each personality package, are displayed on the Protocol Configuration screen until you modify them. You can always recall these defaults to the screen by displaying the appropriate parameters menu and pressing CTRL – CLEAR.

8.4 Saving the Layer Setup Screen

You can save a configured Layer Setup screen and load it later so that you don't have to make selections on the screen each time you use the equipment. This is done by configuring the Layer Setup screen and then saving it along with all other screens as part

of a program or object file (see Figure 8-4). (Program and object files are saved and loaded from the File Maintenance screen as explained in Section 15.) When loaded into the unit, a program (or object) file overwrites all screens.

To use a program or object file with layer packages loaded, be sure that you also make those layer packages accessible to the system when the file is loaded. The INTERVIEW 8000 Series always has access to the layer packages since they reside on the hard disk.

To use a program or object file with personality packages, simply load the file in any drive. All drives will be searched, if present, in the following order for necessary

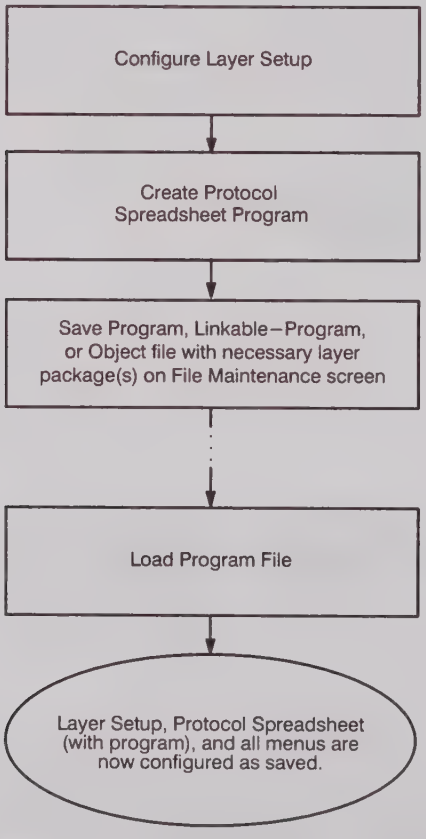


Figure 8-4 Saving and loading the Layer Setup screen.

layer personality packages: boot drive, F1, F2, hard disk. (Of course, the boot drive is F1, F2, or hard disk.) This order is important when unique names have not been used. The first occurrence of the specified protocol package will be used. All protocol packages previously resident in the INTERVIEW will be replaced automatically, as long as the INTERVIEW can access the protocol packages specified in the program file. The personality package need not appear in the same drive from which it was originally loaded.

If the INTERVIEW does not find the Personality Package on any drive, nothing will be loaded. Missing softkey selections and pervasive strike–through's on the Protocol Spreadsheet also indicate that the correct Layer Personality Package has not been loaded.

To correct the problem, reload the program and packages. Return to the File Maintenance screen, insert the user disk containing the personality packages if necessary, and press [XEQ].

9 FEB Setup

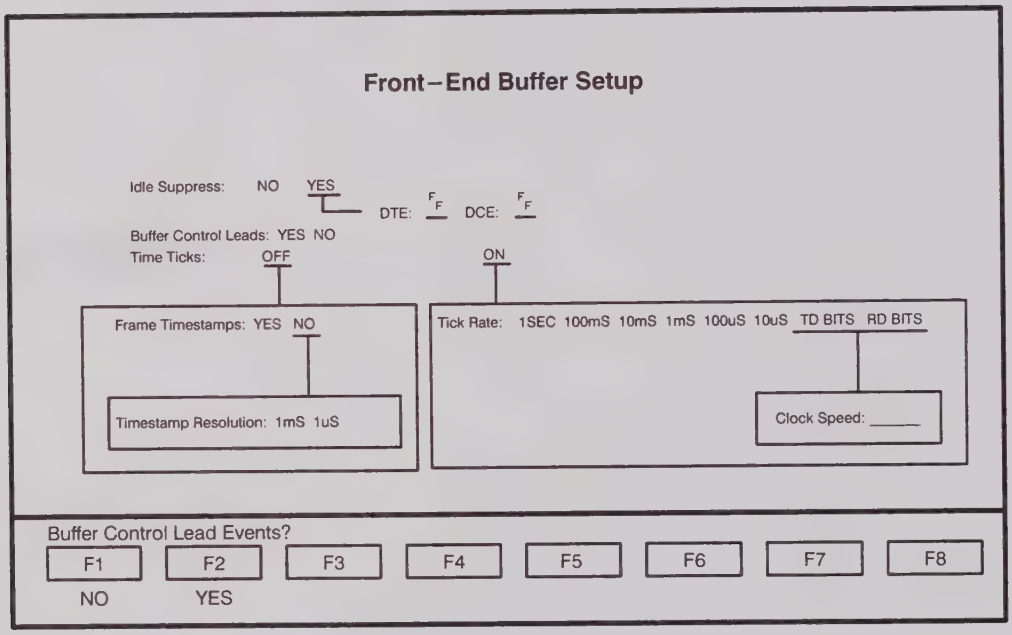


Figure 9-1 Selections on the Front–End Buffer Setup screen.

9 FEB Setup

Data and control–lead signals entering the INTERVIEW or generated internally are routed from the receivers through a front–end buffer (FEB) before being presented to the screen and to the program. Since it holds onto data longer during times of peak processing, the FEB may shorten or lengthen slightly the time interval between signal "events." Buffering will not necessarily affect timer measurements, however, since the FEB has a mechanism that recreates real time to an accuracy of a microsecond for all time–related measurements that the unit performs.

The mechanism of time–recreation is the time tick, counted and encoded in the FEB and decoded during processing. Time ticks are encoded in recorded data whether in bit–image or character format. Very precise timing measurements are available for data in either format when it is played back.

Time ticks are enabled/disabled and their rate selected on the Front–End Buffer Setup screen. Figure 9-1 is an overview of this menu.

An alternate timing mechanism for character data is the frame timestamp. It's value is based on an internal "wall" clock. Frame timestamps require less overhead than time ticks. These timestamps are incorporated at the time a frame's BCC is detected and are only recorded in character mode recordings.

9.1 Buffering Idle, Control Leads, Ticks, and Timestamps

Data bits are buffered automatically in the FEB. The buffering of other events—control leads, idle bits, and time ticks (or timestamps)—can be enabled or disabled on the Front–End Buffer Setup menu.

Suppressing events in the FEB means intercepting them directly from the line so that they are not passed to the screen or to the program. Suppressing control leads means, for example, that neither EIA trigger conditions nor the data–plus–leads display will be available in real time or playback. Suppressing idle means that idle characters are not displayed even when **Display Idle:** ON is selected on the Line Setup menu. Suppressing time ticks means that timers lose some precision, especially when recorded data is being played back. The advantage of these suppressions is that when the user does not care about EIA leads or idle characters or playback timings, they can be dispensed with, in order to save processing time and also memory space in RAM or on disk.

(A) Suppressing Idle

Idle characters from the line can be suppressed in the FEB before they consume valuable space in RAM and on disk. Figure 9-2 shows the selection subfields that allow suppression of separate idle characters for DTE and DCE, as, for example, when a host idles ^E while multidropped terminals are idling ^F. If **Idle Suppress:** YES is selected on the Front–End Buffer Setup screen, nothing can be done in the **Display Idle** field on the Line Setup menu to restore the lost idle characters.

Once recorded, idle characters are locked into the data. Since the hardware that suppresses idle is not on the playback path for disk data, the **Idle Suppress** selection on the FEB setup does not apply. You can still suppress idle from the display via the **Display Idle:** OFF selection on the Line Setup menu or the **Suppress** field on the Display Setup menu.

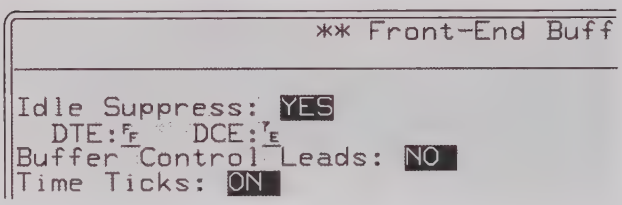


Figure 9-2 The FEB menu allows suppression of separate idle characters for DTE and DCE.

Although they are expensive to record, idle characters can be useful, chiefly to preserve idle time in between data transmissions when recorded data is sent to the screen display. Idle time also is preserved in time ticks if they are enabled; but while they do drive timers, these ticks do not drive the screen display. Figure 9-3 illustrates how idle time may be used to advantage on the screen.

In this figure, the data sample on the top was recorded with idle suppressed, the bottom sample with idle buffered. The time between RTS *on* and CTS *on* was the same in both samples, but only the display that included idle characters retains a picture of the handshake interval.

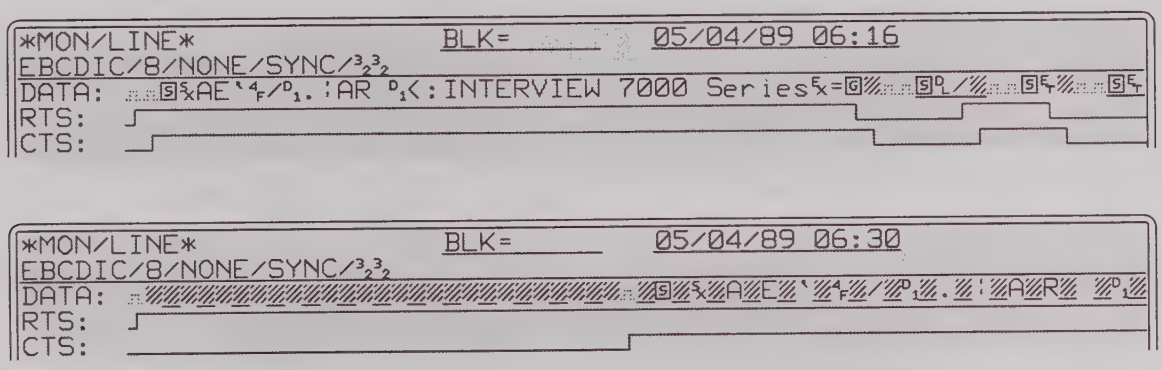


Figure 9-3 In the lower example, idle has *not* been suppressed in the FEB or in the line setup. Note that the interval between RTS *on* and CTS *on* is clearly preserved.

(B) Buffering Control Leads

The next field on the Front–End Buffer Setup screen is **Buffer Control Leads:** NO or YES. See Figure 9-4. Buffering control leads means that control–lead status will be available for data–plus–leads display and triggering. See Section 6.3(B) for a description of the data–plus leads display mode. Data–plus–leads will also be available during playback of bit–image or character data, whether or not this mode of display was used when the data was first recorded.

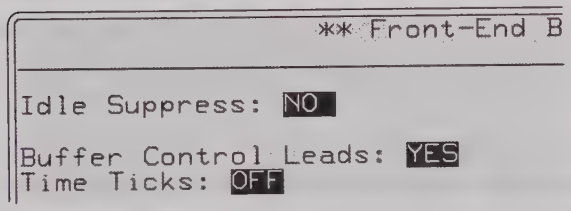


Figure 9-4 Control–lead status, time ticks, and idle characters can be buffered or suppressed in the FEB.

During data recording, a **Buffer Control Leads:** YES selection means that control leads will be recorded with the data. Once control leads are recorded alongside *character* data, they are locked in. Since the FEB is not on the playback path for character data, FEB selections do not apply during playback. (See Figure 2-6.) *Bit–image* data, however, does pass through the FEB during playback. Except for the **Idle Suppress** and **Frame Timestamps** fields, FEB selections apply. This means that control leads *must* be enabled in order for the program logic to detect them.

This field does not affect the front–panel green–red LEDs, which are *always* active for line data and *never* active when recorded data is played back. (If the LEDs are active during playback, they are reflecting line activity and not the data that is being played back.)

(C) Time Ticks in Relation to Timer Units

Another field on the menu allows you to turn time ticks on or off. When **Time Ticks:** ON is selected, time values are incorporated *into the data itself* in the front–end buffer. As a result, internal time measurements such as programmable timer readings, **TIME**–column values on the protocol–trace screens, and so forth, will not be affected when recorded data is played back, even at varying speeds.

> **NOTE:** Once time ticks are recorded alongside *character* data, they are locked in. Since the FEB is not on the playback path for character data, FEB selections do not apply during playback. (Refer again to Figure 2-6.) *Bit–image* data, however, does pass through the FEB during playback. Except for the **Idle Suppress** and **Frame Timestamps** fields, FEB selections apply. This means that time ticks *must* be enabled in order for the program logic to detect them.

> **NOTE:** Since time ticks 1 second to 10 microseconds are added to the recorded data, they are only valid up to 768 Kbps. Operation above these speeds use clock speed or frame timestamp selections. See subsections (D) and (E) below.

If **Time Ticks:** OFF is selected, another field, **Frame Timestamps**, appears on the FEB Setup menu. This field defaults to YES. See Section (E) below for more information on this field.

When **Time Ticks:** ON is selected, a **Tick Rate** field appears just below it with eight selections: 1SEC, 100mS, 10mS, 1mS, 100uS, 10uS, TD BITS, and RD BITS. Tick rate is the interval between ticks. This interval is the smallest unit of measurement attainable by the INTERVIEW's timers. TD BITS and RD BITS are recommended for use with BOP data only; when TD BITS or RD BITS are selected, an additional field, **Clock Speed**, appears. See Section (D) below for more information on this field.

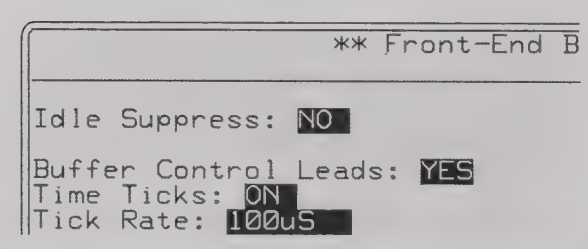


Figure 9-5 The user specifies the degree of timing precision by selecting the *tick rate*, the intervals at which time–ticks are stored with the data.

Ticks are indivisible. A **Unit** of time selected on a statistics screen should not be smaller than the tick rate on the FEB–setup screen. Figure 9-6 shows a mismatch between units on one screen and ticks on the other. **Unit:** MILLI-SECS is selected for a timer on the Tabular Statistics screen; but because the **Tick Rate** entry on the FEB–setup screen is 1SEC, the timer will advance in units no smaller than one thousand milliseconds (or one second). SECONDS would have been a more appropriate unit for this timer.

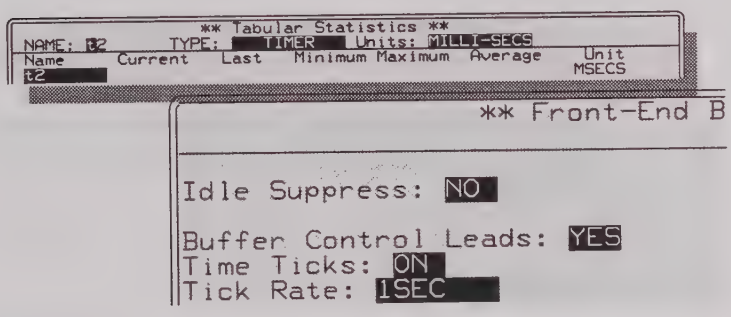


Figure 9-6 This is a mismatch: the **Unit** of time on the statistics screen should not be smaller than the tick rate on the FEB screen.

(D) Clock Speed

This field only appears when **Tick Rate:** TD BITS or **Tick Rate:** RD BITS is selected (see Figure 9-7). The user enters the speed for the TD or RD bit clock selected; however, be aware that this field is overridden by any entry for internal clock rate entered on the Line Setup screen.

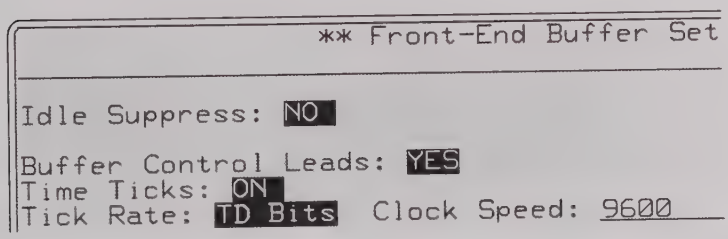


Figure 9-7 The user can specify which bit–clock to use, TD or RD, and its rate in the *tick rate* and *clock speed* fields to control the intervals at which time–ticks are stored with the data.

With this method, the time values are not incorporated directly into the data itself in the front–end buffer, but are calculated by counting the incoming bits for the side of the line selected and factoring it by the entered clock rate. As a result, this allows operation at full speed with no overhead in the recorded data (i.e., 2.048 Mbps for SCSI equipped units). The clock rate information is stored with the recording, and

the time information is not affected by the playback speed. This is only valid when the clock is constant over a period of time and no data is suppressed from the counter (i.e., **Idle Suppress** cannot be functional on the same side of the line).

> **NOTE:** Suppressing idle can be used for certain setups (such as BOP protocols). Since the BOP protocol does bit stuffing for transparency, it never allows the line to contain $^{F}_{F}$. Under these conditions, use the following setup:
>
> **Idle Suppress:** YES
>
> **DTE:** $^{7}_{E}$ **DCE:** $^{F}_{F}$
>
> **Tick Rate:** RD BITS
>
> This is a valid setup for links that idle flags between frames and would conserve recording space while still providing accurate time information.

(E) Frame Timestamps

This field only appears when **Time Ticks:** OFF is selected. The default selection for the **Frame Timestamps** field is YES. This means that time values will incorporated into recorded data at the time that a frame's BCC is detected.

Timing measurements are based on an internal "wall" clock in the INTERVIEW. The measurements can be noted to the millisecond or to the microsecond; select 1mS or 1uS for the **Timestamp Resolution** field, respectively. The **Timestamp Resolution** field only appears when **Time Ticks:** OFF and **Frame Timestamps:** YES are selected.

Frame timestamps may be incorporated into character data only, not bit–image data. Data format is selected in the **Data to Record** field on the Record Setup menu. See Section 7.5(A).

Time ticks are not available for some hardware configurations—e.g., when the T1 or G.703 option is installed. Frame timestamps or Clock Speed—see subsection (D)—provide a method for incorporating timing measurements into received data.

If you select **Frame Timestamps:** NO, time values will not be incorporated into the data. During playback, such timings will be influenced by "local conditions" such as playback speed, idle suppression (on the Display Setup menu), etc.

> **NOTE:** The "clock on the wall" is the timing mechanism used in other older INTERVIEWs such as the COMSTATE series and the 4600, and many users will feel comfortable in turning time ticks off and relying on the wall clock for all normal operations.

10 Block Checking

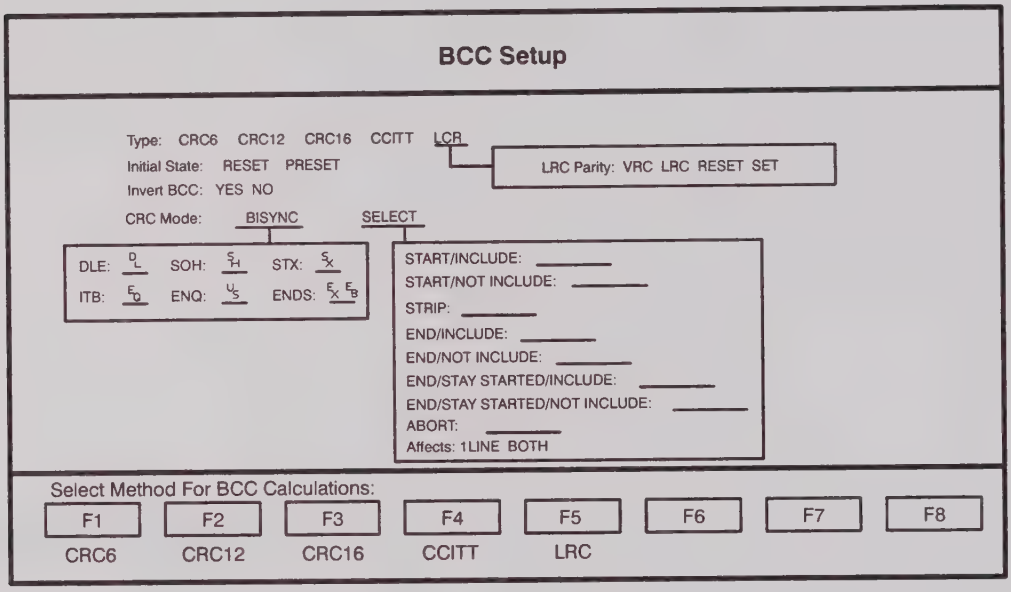


Figure 10-1 Fields and selections on BCC Setup screen.

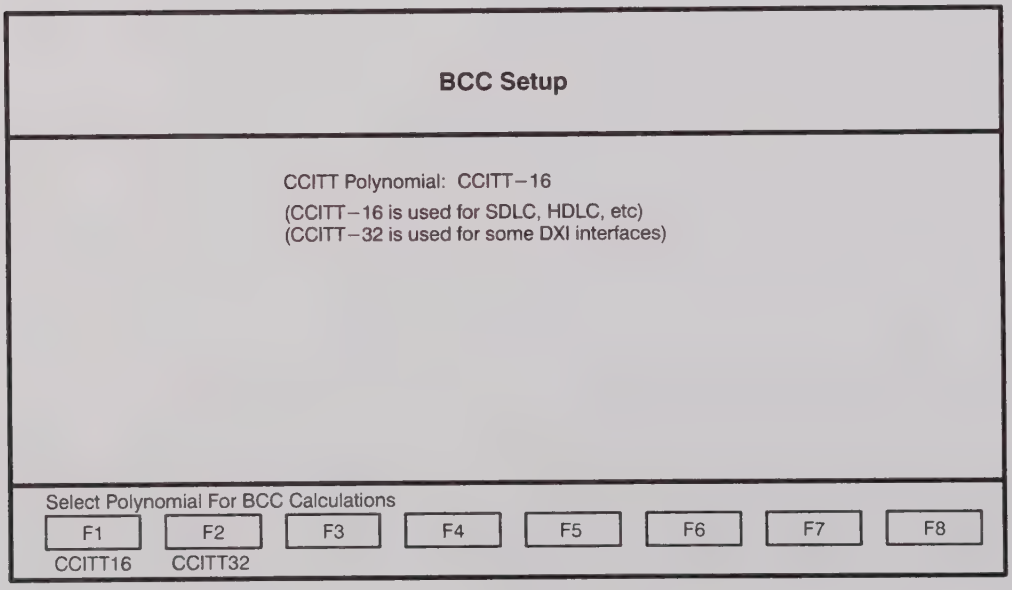


Figure 10-2 Alternate BCC Setup screen in BOP format in High Speed Frame Mode selected or in SYNC format in SMDS Frame Mode.

10 Block Checking

The INTERVIEW is capable of a variety of standard and nonstandard block–check calculations (BCCs). These calculations can be appended to the INTERVIEW's own transmissions and they also can be used for comparison with BCCs in the line data. The results of the comparison are displayed on the unit's monitor as special symbols representing good and bad BCCs. (A "good" BCC in the line data is one that agrees with the INTERVIEW's own internal calculation.) The result of a BCC evaluation can also be used to satisfy a trigger condition.

10.1 BCC Symbols

The internal BCC that the INTERVIEW compares with a BCC in the line data and then displays as a special symbol on the data screen, is enabled in a field on the Line Setup menu. This field is named **Rcv Blk Chk** and is shown in Figure 10-3. When **Rcv Blk Chk:** ON is selected, the unit evaluates as "good" or "bad" the BCCs in all properly framed data blocks. The last byte in the data BCC is then overwritten on the INTERVIEW monitor with G or B. Figure 10-4 shows a BCC symbol written over the second character of a line BCC that has been judged bad.

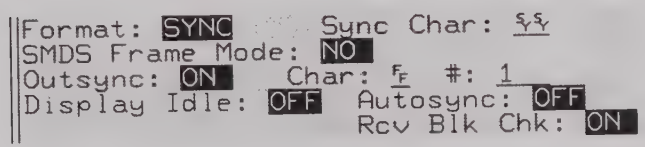


Figure 10-3 A field on the Line Setup menu enables block checking on all data "received" to screen and triggers.

It should be noted that the BCC–appending function and the BCC–evaluating function are separate, and that the **Rcv Blk Chk** field enables BCC *evaluation* in all data, including the INTERVIEW's own transmissions. The data block in Figure 10-4 was transmitted by the INTERVIEW. It is simply a fox message framed by ␗x and ␗x and preceded by sync characters. The transmit trigger that generated the message specified "no BCC" to be appended to the data (see Section 10.3), so ␗x is followed by idle ▨ (pad) characters. But since **Rcv Blk Chk** is enabled, the unit treats the first two idle characters as BCC and evaluates them accordingly.

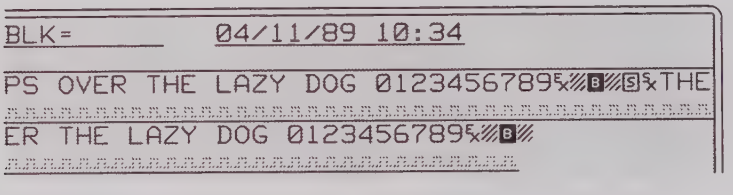


Figure 10-4 A special symbol overlays the final character in a bad BCC.

For non–BOP data, there are two ways to look underneath the G or B overlay to get a glimpse of the concealed final block–check character. One way is to select **Rcv Blk Chk:** OFF . Then the complete block–check calculation monitored by the INTERVIEW is displayed on the screen, with no special symbol overlaying the final BCC character.

The second way to uncover the hidden block–check character is to look at the binary expansion of the character in Freeze mode. In Figure 10-5, the operator has pressed FREEZE and then moved the cursor over the good–BCC symbol. In Freeze mode, the binary expansion of any character that comes under the cursor (including a concealed block–check character) is given at the top right of the screen.

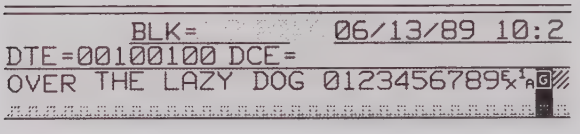


Figure 10-5 From the Freeze–mode binary expansion of the cursor character, it is easy to tell that the the good–BCC overlay is covering a hexadecimal *24*.

For BOP formats, **Rcv Blk Chk** does not appear on the Line Setup menu. Block–check evaluation is automatically *on* for BOP.

In BOP format only, block–check evaluation includes an abort symbol, A. Bisync devices signal an abort by sending ESC in the middle of a text block, but BOP devices send no such control character—they merely idle mark for seven bit–times to indicate an aborted frame. The INTERVIEW uses the A symbol to stamp these seven consecutive 1–bits clearly as an abort.

Control over the type of block check and the many parameters associated with the type is afforded on the BCC Setup screen. The selections on this screen are laid out in Figure 10-1.

The BCC Setup screen does not operate in the same manner for BOP or for SYNC in SMDS Frame Mode (Switched Multimegabit Data Service). BOP format has a well–defined block–check sequence that is not alterable in the INTERVIEW with one exception: the CCITT Polynomial CCITT-16 or CCITT-32 is selectable on the BCC Setup when in High Speed Frame Mode in BOP format—see Section 10.4(A).

10.2 BCC Conditions

Good and bad BCCs (and aborts in BOP format) can be used as trigger conditions. Figure 10-6 shows a GD BCC condition on a trigger menu. And here is an example of a BDBCC condition in a Protocol Spreadsheet test for Layer 1:

```
LAYER: 1
   TEST: bcc
      STATE: bad_bcc
         CONDITIONS: RECEIVE BAD_BCC
         ACTIONS: COUNTER bad_bcc INC
```

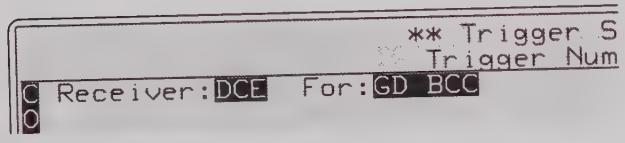


Figure 10-6 Good and bad BCCs can be used as trigger conditions.

Figure 10-7 is a string condition with a further condition added: Wait for E(nd) O(f) F(rame), the literal meaning of which is "wait for a good BCC." None of the triggers in these examples can come true unless block–check evaluations are enabled on the Line Setup menu. This enabling is automatic in BOP format. In other formats, **Rcv Blk Chk**: ON must be selected as a line–setup parameter.

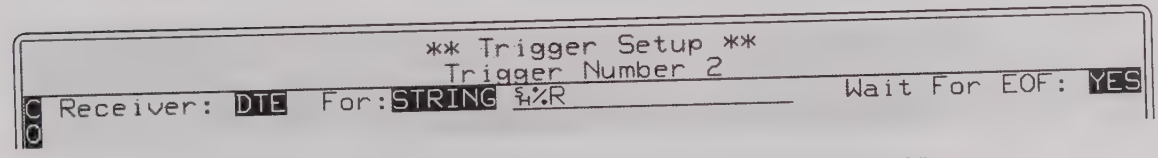


Figure 10-7 *Wait For EOF* following a *Receiver* condition means "wait for a good BCC."

10.3 Transmitted BCC

Block–check calculations that the INTERVIEW appends to messages and transmits out onto the line are enabled in the **BCC** field on trigger menus and in similar entries on the spreadsheet. **BCC** is a subfield under **Xmit** on trigger–action menus (see Figure 10-8). On the spreadsheet, transmitted BCC is a subselection under SEND: every time you transmit ("send") a message you have a choice of appending a good BCC, a bad one, an abort (BOP only), or nothing (not applicable to BOP).

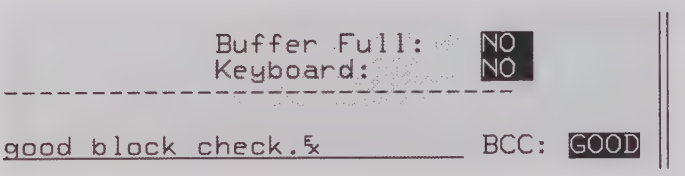


Figure 10-8 A transmitted BCC may be selected on the *actions* half of a trigger menu.

Please note that **BCC:** GOOD or BAD following a transmit string does not mean that a BCC will always follow. In bisync, the INTERVIEW will append BCCs only to text strings that are *properly framed*—as, for example, by STX and ETX (or ETB). If you are sending a bisync poll or ACK or ENQ or some other nontext message, your selection in the **BCC** field will have no significance.

BCC: NONE means that the INTERVIEW will go directly from transmitting the transmit string to idling mark, even if the string begins with STX and ends with ETX or ETB. The unit's receivers, if they are enabled for block checking and if they stay in sync for the requisite number of BCC characters, will interpret this no–BCC as a bad BCC (Figure 10-4).

In BOP format, the sudden shift from data to mark idle is an abort. The third BCC selection in BOP triggers therefore is called ABORT instead of NONE. The receivers put up the appropriate ▣ symbol when they see the seventh mark bit.

Good BCCs are transmitted in accordance with the parameters in effect on the BCC Setup menu. See Section 10.5.

10.4 Standard BCC Parameters

A specific set of block–check parameters is standard for each code selected on the Line Setup menu with two exceptions—BOP in High Speed Frame Mode and SYNC in SMDS Frame Mode. These exceptions are explained in Sections 10.4(A) and (B).

Table 10-1 summarizes the correct BCC settings for the various other standard codes and formats supported by the INTERVIEW.

The actual equations used by the INTERVIEW in block–check calculations are defined in Table 10-2.

Changes to the BCC Setup menu remain on the menu when you change line–setup formats.

Table 10-1
Standard Block–Check Parameters for Sync[1] or Async/Isoc Formats

Code	Type	BCC starts following	BCC aborts/resets on	BCC ends with
EBCDIC	CRC–16	$^{5}_{H}$ or $^{5}_{X}$	$^{E}_{0}$ or next bit after BCC	$^{E}_{B}$, $^{E}_{X}$
ASCII	LRC	$^{5}_{H}$ or $^{5}_{X}$	$^{E}_{0}$ or next bit after BCC	$^{E}_{B}$, $^{E}_{X}$
IPARS	CRC–6	SY2 ($^{3}_{E}$)	SY1 ($^{3}_{F}$) or next bit after CRC	EOM–PB ($^{3}_{D}$) EOM–1 ($^{0}_{D}$) EOM–C ($^{1}_{D}$) EOM–U ($^{2}_{D}$)
BAUDOT	LRC	––	––	––
EBCD/ SELECTRIC	LRC; *Xmit:* LRC parity=LRC *Recv:* ignores parity bit	*Xmit:* sync *Recv:* bid (#)	$^{E}_{T}$ on either side resets BCC for both sides	$^{E}_{B}$
XS–3	LRC	$^{4}_{0}$	––	$^{5}_{S}$ (not included in BCC)

1 Not in SMDS Frame Mode

Table 10-2
Block Check Polynomials

BCC Type	Block Check Polynomial
CRC–32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
CRC–16	$X^{16}+X^{15}+X^{2}+1$
CRC–CCITT	$X^{16}+X^{12}+X^{5}+1$
CRC–12	$X^{12}+X^{11}+X^{3}+X^{2}+X+1$
CRC–8	$X^{8}+X^{2}+X+1$
CRC–6	$X^{6}+X^{5}+1$

(A) BOP Format

For Bit–Oriented Protocols, the INTERVIEW uses BCC calculations standard for BOP; when the unit is in High Speed Frame Mode, the user can select the CCITT polynomial.

1. *Not In High Speed Frame Mode.* When **High Speed Frame Mode:** NO is selected on the Line Setup screen, the BCC Setup screen is not accessible. The unit uses BCC calculations standard for BOP. Instead, Figure 10-9 is displayed.

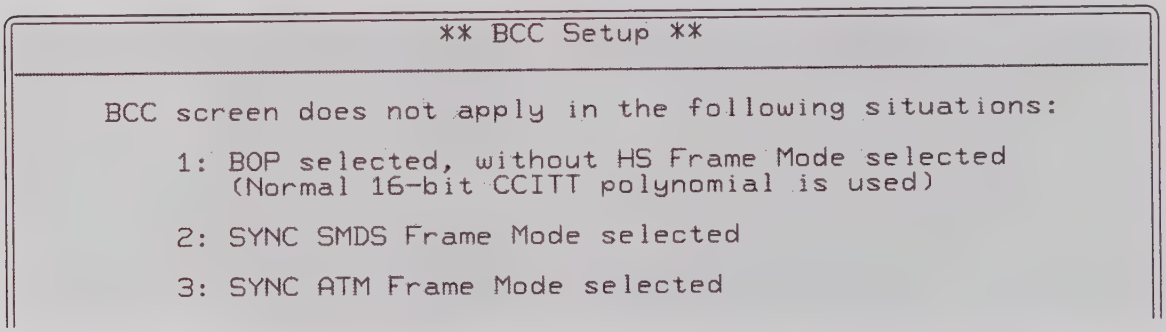


Figure 10-9 BCC Setup display in BOP format with High Speed Frame Mode selected or in SYNC format with **HS Frame Mode:** SMDS or ATM selected.

2. *In High Speed Frame Mode.* When **High Speed Frame Mode:** YES is selected on the Line Setup screen, the unit uses standard parameters for BCC calculations with one user–selectable field: the user may select the **CCITT Polynomial:** CCITT-16 or CCITT-32 on the BCC Setup screen as shown in Figure 10-10.

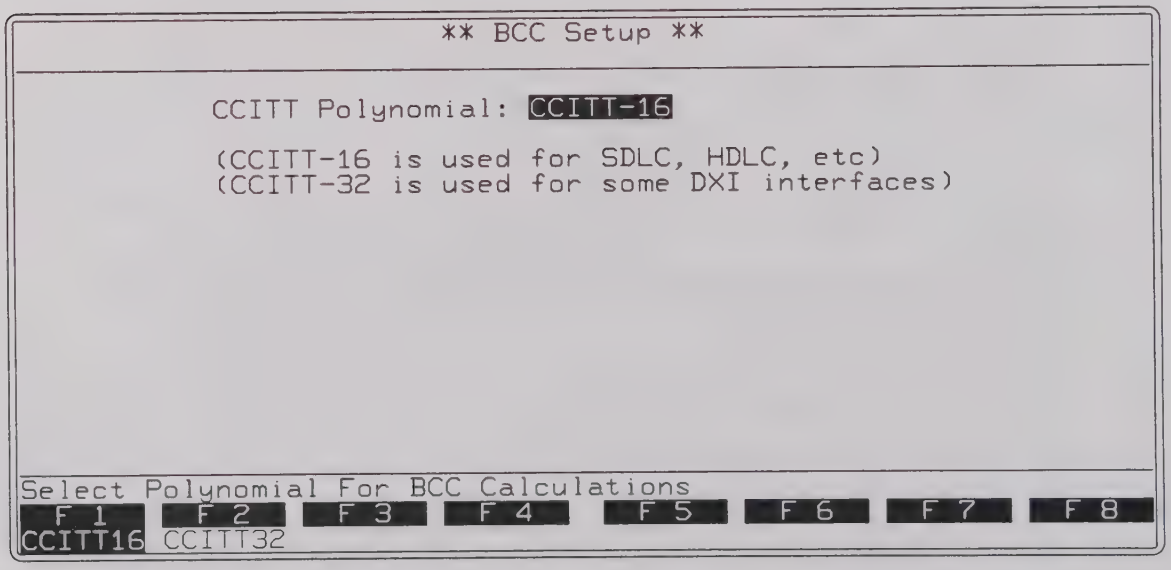


Figure 10-10 The only BCC parameter selection for BOP in High Speed Frame Mode is the CCITT polynomial.

In this mode an INTERVIEW 8800 *TURBO* supports applications for monitoring and emulation of the SMDS DXI (Data Exchange Interface) at T1 and E1 data rates with no additional hardware requirement. The DXI protocol is used between routers and SMDS DSUs in BOP format with an optional 32−bit CRC; select **CCITT Polynomial:** CCITT−32 on the BCC Setup screen as shown in Figure 10-10.

(B) Sync and HS Frame Mode Formats

For Sync in SMDS Frame Mode (Switched Multimegabit Data Service) and ATM Frame Mode (Asynchronous Transfer Mode), the INTERVIEW defaults to the BCC calculations standard for these modes. The parameters for these calculations cannot be changed on the BCC Setup screen and do not appear—the standard BCC Setup screen is inaccessible (see Figure 10-9).

(C) Sync and Start−Stop Formats

When a format other than BOP is selected on the Line Setup menu, the unit must be set up for BISYNC or SELECTABLE **CRC Mode** depending on the code selected on the Line Setup menu. The various correct configurations for SYNC or start−stop formats (ASYNC and ISOC) are detailed below.

1. *EBCDIC or ASCII code*. Both EBCDIC and ASCII require **CRC Mode:** BISYNC, but EBCDIC uses a 16−bit CRC−16 calculation while ASCII uses an 8−bit LRC check. You may redefine the values of the BCC control characters using alternate control characters, standard alphanumerics, or hexadecimal characters. You may also indicate that the control character does not exist by leaving the field blank. See Section 10.6 on bisync CRC−mode operation.

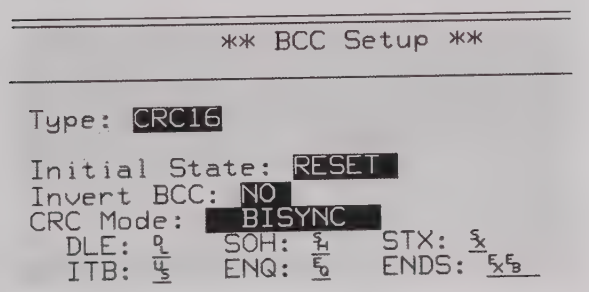


Figure 10-11 BCC setup for EBCDIC.

2. *IPARS*. Selecting **Code:** IPARS in sync or async (or isoc) format requires the BCC parameters shown in Figure 10-13. IPARS requires **CRC Mode:** SELECTABLE and uses CRC6 to calculate BCCs. Any of the BCC parameters may be changed to meet specific applications.

3. *BAUDOT.* Since Baudot does not normally support block checking, there are no standard settings.

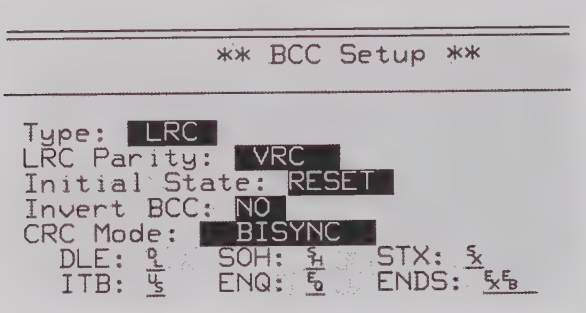


Figure 10-12 BCC setup for ASCII.

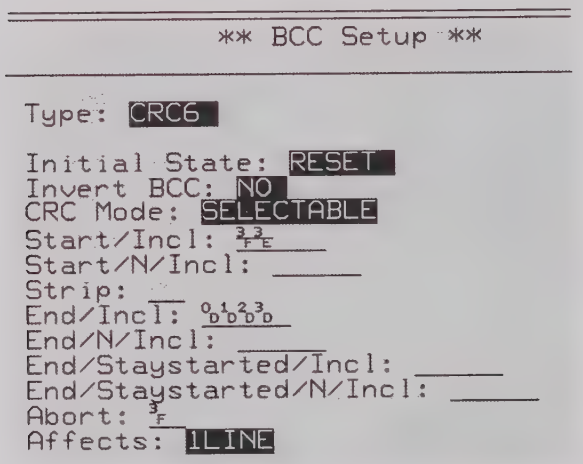


Figure 10-13 BCC setup for IPARS.

4. *EBCD, XS−3, or SELECTRIC.* Selecting EBCD or SELECTRIC code for SYNC or for either of the start−stop formats will require the BCC parameters shown in Figure 10-14. The XS_3 parameters are shown in Figure 10-15. All three codes require SELECTABLE CRC mode and use an LRC check to determine BCCs. Any of the BCC parameters may be changed to meet specific applications.

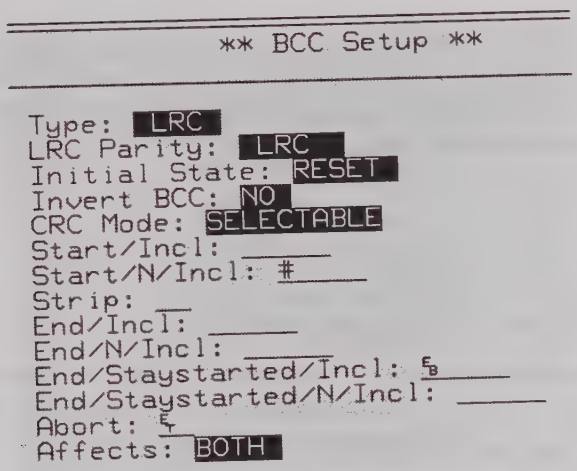


Figure 10-14 BCC setup for EBCD or SELECTRIC.

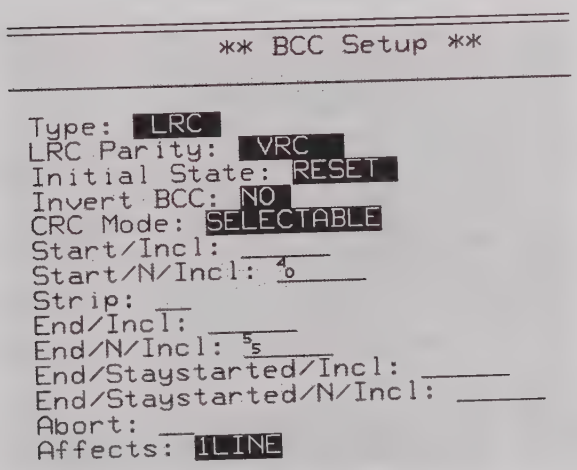


Figure 10-15 BCC setup for XS−3.

10.5 BCC Setup Menu Fields

The BCC Setup Menu controls and displays the values of the INTERVIEW's block check parameters. The full set of parameters is shown in Figure 10-1. The meanings of the BCC Setup parameter fields are found in Table 10-3.

Entries on the menu may be made in either alphanumerics, control characters, or hexadecimals. In the **START/INCL** field only, characters may also be entered in the not−equal format. See, for example, the IPARS setup in Figure 10-13.

Use control characters instead of hexadecimals where possible, since hexadecimals commit you to a particular parity that may change later on. An ASCII ENQ, for example, is hex 05_2 or 85_2 depending on the parity selected on the Line Setup menu. If you enter 05_2 in the **STX** or **START/N/INCL** field on the BCC Setup screen, the software will recognize ENQ only in odd–parity ASCII data. An entry of ENQ, on the other hand, will adjust for whatever parity is enabled on the Line Setup menu.

Table 10-3
BCC Setup Menu Fields

Type	Indicates method of BCC calculation selected. Polynomial expansions of each CRC type are listed in Table 10-2.
LRC Parity	Displayed when *Type: LRC* is selected. Identifies how the parity bit in the BCC character is calculated. *LRC* = parity bit in BCC character the result of an LRC on the parity bits within the message *VRC* = parity bit the result of a VRC on the BCC character *RESET* = parity bit always 0 *SET* = parity bit always 1
Initial State	Sets initial state of block check character. *RESET* = all 0's. When *Type* is *LRC*, this selection yields an *even* longitudinal check. *PRESET* = all 1's. When *Type* is *LRC*, this selection yields an *odd* longitudinal check.
Invert BCC	*YES* produces an inverted BCC by changing 1's to 0 and 0's to 1.
CRC Mode	Allows choice between *BISYNC* and *SELECTABLE* CRC modes.
BISYNC control character fields	Data–entry fields displayed for *BISYNC* CRC mode only. Allow you to select the characters that control block–checking. Default is standard set of bisync control characters. Alphanumeric, hexadecimal, and control characters are legal.
START/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW initiates BCC accumulation, and includes the character(s) in the accumulation. For this field only, characters may be entered in either normal or not–equal format.

Table 10-3 (continued)

START/N/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW initiates BCC accumulation, and does not include the character(s) in the accumulation.
STRIP	Displayed when *SELECTABLE* is chosen. Identifies character(s) to be stripped from BCC accumulation.
END/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW ends BCC accumulation, and includes the character(s) in the accumulation. Initiates processing of BCC. Returns to START state when processing is complete (see Figure 10-17).
END/N/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW ends BCC accumulation, and does not include the character(s) in the accumulation. Initiates processing of BCC. Returns to START state when processing is complete (see Figure 10-17).
END/STAYSTARTED/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW ends BCC accumulation, and includes the character(s) in the accumulation. Initiates processing of BCC. Returns to ACCUMULATE state when processing is complete (see Figure 10-17). This function is performed by the ⅄ (Intermediate Block–check or ITB) character in bisync.
END/STAYSTARTED/N/INCL	Displayed when *SELECTABLE* is chosen. Identifies the character(s) on which the INTERVIEW ends BCC accumulation, but does not include the character(s) in the accumulation. Initiates processing of BCC. Returns to ACCUMULATE state when processing is complete (see Figure 10-17).
ABORT	Displayed when *SELECTABLE* is chosen. Identifies character(s) on which the INTERVIEW aborts BCC accumulation and returns to START state (See Figure 10-17). This function is performed by ⅇ in bisync. Note that the abort *function* does not generate an abort *overlay* on the screen. The user may enhance the abort character in the *Enhance* field on the Display Setup menu. See Section 6.3(D).
Affects	This field pertains to the *ABORT* character on the preceding line. The choices are *1LINE* or *BOTH*. The abort character may cause only the side of the line sending the character to reset its BCC; or it may have this affect on both sides. ⅂ in EBCD/SELECTRIC is an example of a character that resets BCC on both sides.

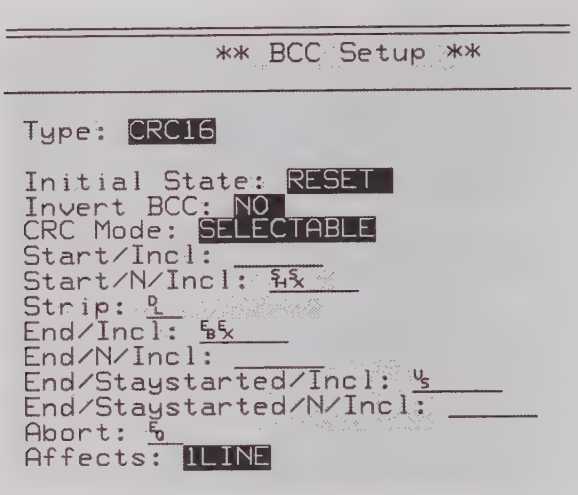


Figure 10-16 A valid BCC for nontransparent bisync can be configured "manually" using *SELECTABLE* parameters.

10.6 BISYNC vs. Selectable CRC Mode

The INTERVIEW supports an expanded subset of IBM's "Binary Synchronous Control Procedures" (BISYNC) that covers a wide variety of BISYNC−type applications including Burroughs, Honeywell, Univac, ISO−1175, and others.

Figure 10-11 shows the menu subfields under **CRC Mode:** BISYNC . In these subfields you may specify the values of the six BISYNC control characters (DLE, ITB, SOH, ENQ, STX, and ENDS) which appear on the BCC Setup menu with default character values already assigned (see Figure 10-11).

The BISYNC configuration has one important advantage, however, in that it implements full ^L transparency. When **CRC Mode:** BISYNC is employed, the appearance of ^L^X will enable a transparent mode in which all control characters except ^L are accumulated as data characters while their control functions are ignored. SELECTABLE mode offers no similar mechanism for treating control characters as data. The SELECTABLE setup in Figure 10-16 will work, therefore, only as long as the bisync data is not transparent.

Most of the bisync control characters have counterparts on the *other* set of BCC parameters, accessed under **CRC Mode:** SELECTABLE . Figure 10-16 shows how a standard bisync BCC would look if it were configured using the SELECTABLE subfields on the BCC Setup screen. Compare this screen with the bisync screen in Figure 10-11. Note that the names of the fields in Figure 10-16 are functionally descriptive. **S**tart−**O**f−**H**eader and **S**tart−of−**TeX**t characters, for example, are really **START/N**ot/**INCL**uded characters. This indicates that they activate block−check accumulation but they are not themselves included in the calculation. E_B and E_X, on the other hand, which end the BCC accumulation, *are* included in the block−check calculation. Note also that the set of SELECTABLE parameters is more complete, and that thirty−four distinct characters may be designated as control characters compared to eight in the BISYNC fields.

A state diagram of SELECTABLE parameters is presented in Figure 10-17. In this diagram, process−BCC state does the following:

- Calculates and displays BCC result.
- Reinitializes BCC remainder.
- If end character was a STAYSTARTED character, returns to accumulate state.
- Otherwise, returns to start state.

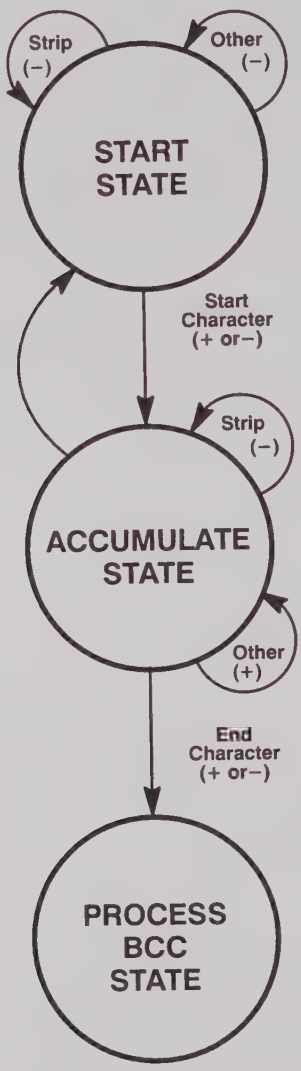


Figure 10-17 State diagram for **CRC Mode:** *selectable*. (+) means the character is included in BCC calculation; (–) means the character is not included.

11 Bit Error Rate Testing

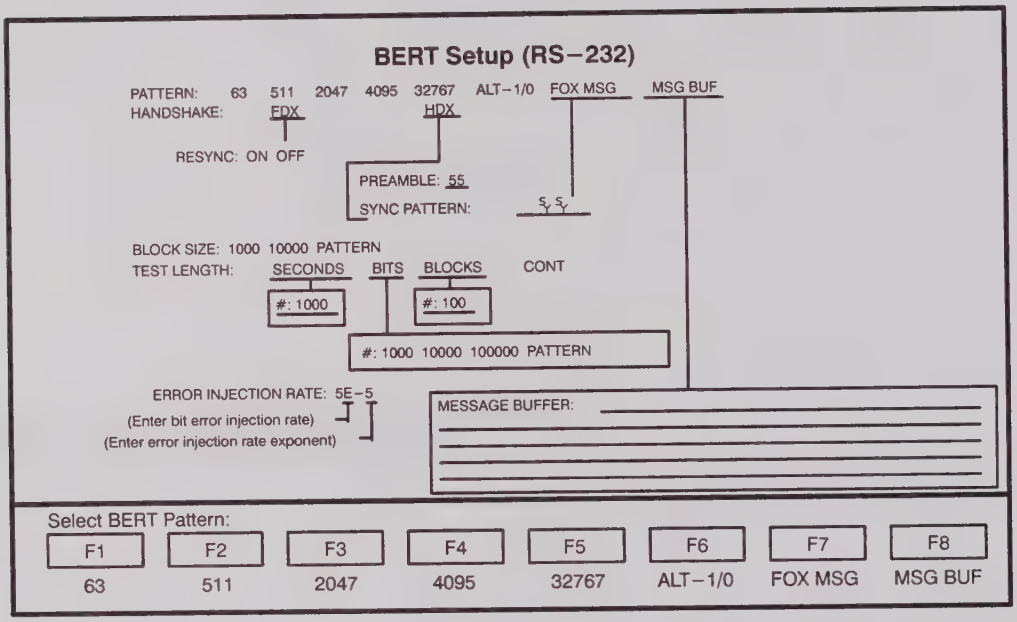


Figure 11-1 Menu selections on the BERT Setup screen.

11 Bit Error Rate Testing

The INTERVIEW can transmit and analyze Bit Error Rate Tests consisting of five different pseudorandom bit patterns, a series of alternating 1's and 0's, a canned fox message, and a user–assigned message of up to 259 characters. The INTERVIEW can send and analyze BERT patterns in synchronous or asynchronous format over transmission facilities that are full duplex or half duplex.

11.1 Pseudorandom Bit Patterns

BERT data may be transmitted and analyzed in pseudorandom patterns of 63, 511, 2047, 4095, or 32767 bits. The algorithm for each pattern is diagrammed in Figure 11-2.

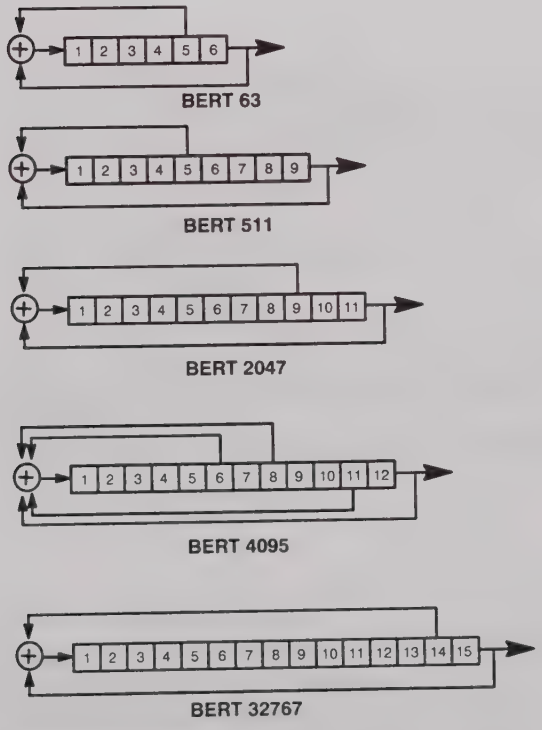


Figure 11-2 BERT algorithms.

11.2 Test Configurations

Tests can be configured for one tester or two.

A standard one–tester configuration places the remote modem (remote from the INTERVIEW) in a loopback condition. See Figure 11-3. The INTERVIEW generates the BERT data and analyzes the data upon its return over the transmission line. This is the easiest test to configure, since the INTERVIEW's BERT analyzer automatically looks for the same pattern of data that the unit generated.

When another BERT generator is used at the remote site, each tester analyzes the data generated by the other. See bottom of Figure 11-3. There is no looping of data. Both testers must generate data that matches bit for bit. Selections on the Line Setup and BERT Setup screens let the user control the pattern of information bits, the number of information bits allotted to a character, the stop bits used along with a start bit to frame each character (async only), and the sync characters necessary for locating (and relocating) the beginning of a fox or user–defined pattern.

The two–tester configuration can be thought of as two separate one–way tests. From the point of view of each tester, the transmitted BERT data is superfluous. The data that is received and analyzed is the test data.

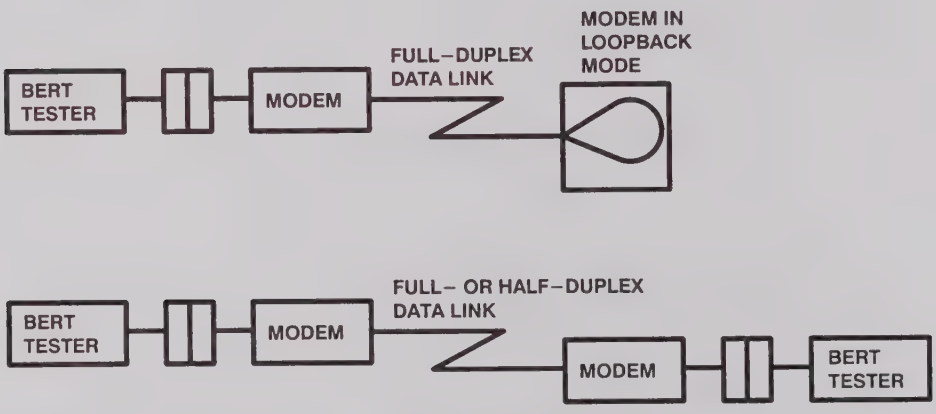


Figure 11-3 Two test configurations: loopback and two–tester.

11.3 BERT Operation: Full Duplex

When the test–interface module (TIM) for RS–232/V.24 is locked in place in the slot at the rear of the unit (see Section 1), the RS–232 subset of parameters on the BERT Setup screen is enabled automatically. On the RS–232 Interface Control menu, **RTS**, **CTS**, and **CD** all should be set to ON for full–duplex operation.

In the **Mode** field on the Line Setup menu, select either BERT DTE or BERT DCE. BERT is used most commonly to test modems and transmission links, so the normal attitude of the INTERVIEW will be **Mode:** BERT DTE. See Figure 11-4.

CAUTION: As soon as the INTERVIEW is run in either of its BERT modes, it will interfere with any active communications on the interface. Be sure that it is all right to break the line and transmit test data on it.

Note on synchronous and BOP operation: The INTERVIEW can transmit using internal clock (SCTE) when it is emulating a DTE; but in order to display its own transmit data as well as receive data, it must be connected to a modem or other DCE device that is providing SCT and SCR clock.

If you have selected **Mode:** BERT DTE and external clock is not available for synchronous or BOP operation, choose **Clock Source:** INTERNAL and patch SCTE to SCT.

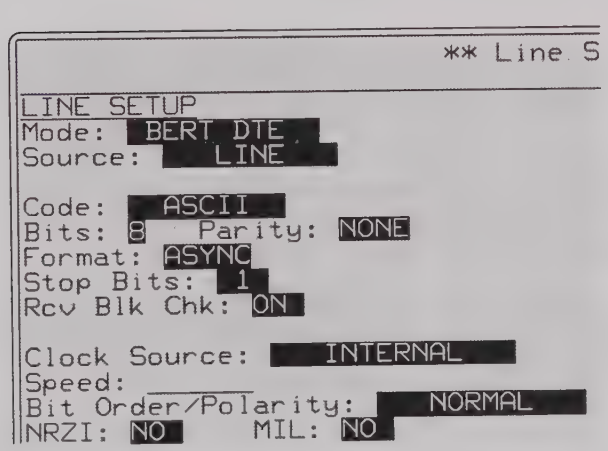


Figure 11-4 The fields on this Line Setup menu—except for *Source*, *Rcv Blk Chk*, and *NRZI*—also are BERT parameters.

Once you have selected the BERT mode, press PROGRAM, F1, F5, to enter the BERT Setup parameters menu (Figure 11-5). Select **Handshake:** FULL DUPLEX in the second field on this menu. When you have completed your other selections in these parameter fields (see Section 11.5), press RUN. In full–duplex testing, whenever you execute a run, the INTERVIEW will begin to operate both as a BERT generator and analyzer.

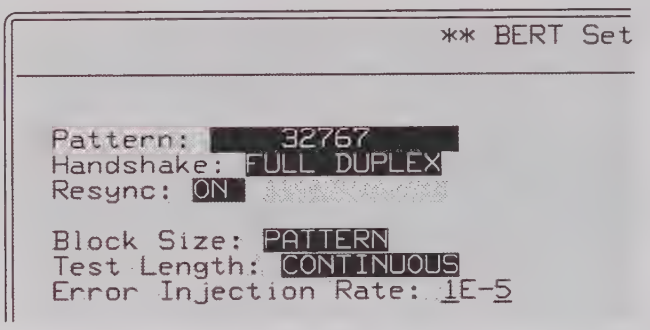


Figure 11-5 BERT Setup screen.

The unit will begin transmitting immediately. The transmission will consist of the bit pattern chosen on the BERT Setup menu according to the format entered on the Line Setup menu.

The INTERVIEW also begins immediately to perform a BERT statistical analysis on all received data. If you press [F3], RESET, while the pattern is being transmitted, the pattern will not be interrupted but the statistical counters will clear. See Section 11.7(E).

11.4 BERT Operation: Half Duplex

To operate over half–duplex transmission lines, go to the RS–232/V.24 Interface Control screen (press [PROGRAM], [F1], [F2]) and change the handshaking control leads (**RTS**, **CTS**, and **CD**) from ON to SWITCH. On the BERT Setup menu, select **Handshake:** HALF DUPLEX.

During the test you will transmit data one block at a time. (See Section 11.5(F) for a definition of "block.") After every block of information that it transmits, the INTERVIEW will drop its control lead and relinquish the data link while the remote tester transmits a block.

On the Line Setup menu, select **Mode:** BERT DTE. Then press [PROGRAM], [F1], [F5], to enter the BERT Setup screen. When you have made your selections in these BERT–menu fields (see Section 11.5), press [RUN].

> *CAUTION: As soon as the INTERVIEW begins running in either of its BERT modes, it will interfere with any active communications on the interface. Be sure that it is all right to break the line and transmit test data on it.*

When the INTERVIEW operates in half duplex, it does not transmit and receive BERT data at the same time. When you send the Run command to start the test, the unit is in the "receive and analyze" mode. Once it has received a complete block of data, it will shift to "generate" mode and transmit *one* block automatically.

In this situation, both ends are waiting; press F5, START, to **initiate** the sending–receiving cycle with a transmission.

11.5 BERT Setup Screen

(A) Pattern

With the cursor in the **Pattern** field, press ROLL FWD and ROLL BACK to rotate between a series of alternating 1's and 0's, a canned fox message, a user–assigned message of up to 259 characters, and five different pseudorandom bit patterns. The INTERVIEW both transmits and expects to receive this pattern while it is in BERT Mode.

When MESSAGE BUFFER is selected in this field, the pattern will consist of the contents of the 259–byte message buffer represented by five data–entry lines toward the bottom of the BERT Setup menu.

Do not run the 1010 pattern if you have selected ASYNChronous (or ISOChronous) start–stop framing on the Line Setup menu.

(B) Handshake

In this field, select **Handshake:** FULL DUPLEX or HALF DUPLEX. The selection will determine whether preamble and sync characters can be appended to the data (half duplex) and whether the resync function can be enabled (full duplex only). In Run mode, the full–duplex pattern will run continuously for the duration of the test, while the half–duplex test will be transmitted one block at a time before turnaround. This field configures the test sequence, not the interface. The interface must be configured on the Interface Control screen in accordance with Table 11-1.

Table 11-1
Full– or Half–Duplex BERT
(RS–232/V.24 Interface Control screen)

	RTS	CTS	CD
Full duplex:	on	on	on
Half duplex:	switch	switch	switch

(C) Resync

Use the resync function to prevent one missed bit–time from skewing an entire test. **Resync:** ON is valid only when the screen is configured for full duplex. This selection allows you to go out of test–sync after a fault has been detected (see Section 11.9(G),

Number Of Faults) and back into synchronization after a short interruption. When out of sync, the receiver stops analyzing bits and counting errors and waits for synchronization to turn the analyzer on again.

Resync: ON avoids a bit–error rate approaching fifty per cent for pseudorandom patterns over a long error count. (The analyzer will "guess" right fifty per cent of the time even if it is out of sync with the incoming test data.)

Table 11-2
BERT Pattern (BERT Setup screen)

PATTERN:	BERT ALGORITHM (63, 511, 2047, 4095, 32767)	ALT–1/0 (synchronous only)	FOX/MSG BUF
Handshake:	Full or half duplex	Full or half duplex	Full or half duplex
Resync on:	Algorithm (full duplex) or sync chars (switched)	1010 (full duplex) or sync chars	one or two sync chars
Preamble char:	selectable (switched line–use only)	selectable (switched line–use only)	selectable (switched line–use only)
Sync chars:	1st pattern–byte selectable, switched line–use (half duplex) only	selectable, switched line–use (half duplex) only	selectable
Block size:	1 Kbit, 10Kbit, or pattern	1 Kbit or 10 Kbit	1 Kbit, 10 Kbit, or pattern
Test length:	selectable # of seconds or blocks; 1 K, 10 K, 100 K, or pattern # of bits; or continuous	selectable # of seconds or blocks; 1 K, 10 K, 100 K, or pattern # of bits; or continuous	selectable # of seconds or blocks; 1 K, 10 K, 100 K, or pattern # of bits; or continuous
Error Injection Rate:	selectable	selectable	selectable

For a pseudorandom pattern, resync means going out of sync and performing the algorithm on two bytes of data in order to predict the third byte. Since the algorithm may be performed at any point in the cycle, there are no sync characters.

For the fox pattern or the user–defined message buffer, resync means going out of sync when a fault occurs and looking bit by bit for the sync pattern entered in the **Sync Pattern** field on the BERT Setup (not the Line Setup) menu.

In half–duplex BERT testing, **Resync:** ON is not available. The analyzer goes out of synchronization at the end of each block when the line turns around, but never in the middle of a block.

A fault in half duplex means that synchronization was missed entirely for a block. The analyzer stays out of sync until the next line turnaround.

On a noisy circuit, a fault may not imply that the analyzer has lost synchronization with the incoming BERT pattern. In spite of a high error rate, the test should remain in sync. Select **Resync:** OFF to prevent the analyzer from going out of sync and suspending its error count while waiting to resynchronize.

(D) Preamble

This selection is enabled when the BERT Setup menu is configured for half duplex only. One or two bytes selectable by the user can be prefixed to the sync pattern in half duplex tests. The default entry in this field is two bytes of ^{5}s, a character with an alternating pattern of 0 and 1 bits. After line turnaround, a modem can use this pattern from a remote DTE to put its bit clock into phase with the new carrier.

Since preamble characters always follow carrier turnaround and precede synchronization, they are not checked by the BERT analyzer for error.

Pressing CLEAR to blank this field will prevent the preamble pattern from being transmitted.

(E) Sync Pattern

Sync characters have a special role in BERT testing. Because they provide message synchronization as well as character sync, BERT analysis requires sync characters in line setups where they would not normally occur—in asynchronous fox–pattern testing, for example. To cover these special applications, the BERT Setup screen is provided with its own **Sync Pattern** field. The **Sync Char** field on the Line Setup menu is inoperative during BERT tests.

1. *Sync characters in fox or user–defined tests*. Default is $^{5}y^{5}y$ in the four–character **Sync Pattern** data–entry field. Alphanumerics, control characters, and hexadecimals are legal. Entry of one, two, three, or four characters is legal. A blank field is treated as default: $^{5}y^{5}y$.

   In full–duplex BERT tests, sync characters precede each fox message or user–defined message. In normal synchronous data transmission, sync characters help the receivers locate character boundaries. In BERT tests they do more: they help the analyzer find message boundaries. For this reason, sync characters precede fox and user messages in asynchronous tests as well as synchronous.

In a full–duplex fox or user test with few errors, the sync pattern is transmitted repeatedly but used only once by the analyzer. After initial synchronization, the analyzer stays in sync and does not look for the sync pattern. The sync characters that precede each successive message in the test are treated as data and checked for bit errors.

In half–duplex BERT, the sync characters precede every transmission in fox and user tests. They are not repeated in mid–transmission. They never are treated as data.

BERT testing can be tricky when two different brands of tester are being used, especially when a data pattern is being tested. Even fox messages will vary with different testers. Some use STX–ETX, some say "JUMPED . . . DOGS" instead of "JUMPS . . . DOG," and so on. Design a MESSAGE BUFFER test to mirror the fox message of the other tester. Use the **Sync Pattern** field for the first character (or more) of the fox message ("%," for example, or "T") in cases where the other tester does not send %%. Then continue the message on the first line of the **Message Buffer**. Figure 11-6 shows how the **Sync Pattern** field is used in conjunction with the message buffer to design a customized fox pattern.

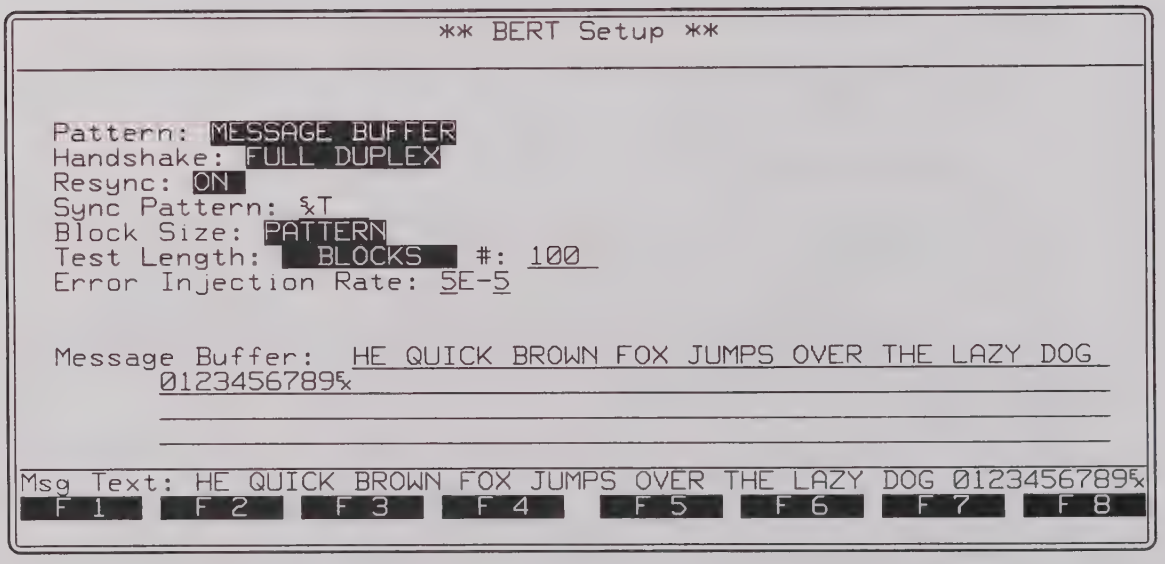


Figure 11-6 If the other tester does not send *sy sy*, synchronize on the first two characters which are sent.

2. *Pattern–synchronization in pseudorandom tests*. For a pseudorandom pattern in full duplex, the **Sync Pattern** field does not apply. Since the algorithm may be performed at any point in the cycle, there are no sync characters.

Synchronization for half−duplex pseudorandom tests is more complicated. The first few bits received after the line turns around are important bits to be tested. Both the transmitting and receiving testers must agree in advance on the point in the algorithmic series where the half−duplex test will begin.

Here the **Sync Pattern** field is used not for standard sync characters but rather to identify the point in the pseudorandom pattern where the test will begin after each line turnaround. This special use of the **Sync Pattern** field allows the operator to program the INTERVIEW for compatibility with half−duplex testers from other manufacturers.

To program a synchronization point into a half−duplex test, the user must know the eight bits that precede the point and the eight bits that follow. For example, suppose that the eight bits preceding a sync point in a 2047−bit test are 10100001.

> **NOTE:** In the notation above, the first bit transmitted is the rightmost. This is consistent with the presentation of binary patterns in all INTERVIEW screen−displays and documents. This right−to−left ordering of bits is well suited to binary−to−hexadecimal and binary−to−ASCII/EBCDIC conversions; but the usual presentation of bits in the BERT literature is left to right.

Since this binary pattern corresponds to the hexadecimal number $^{8}_{1}$, the user would move the cursor to the **Sync Pattern** field and enter $^{8}_{1}$ in the first data−entry position. Suppose also that the eight bits following the sync point (and therefore the first eight bits transmitted after line−turnaround) are 01001000 (hex $^{4}_{8}$). The user enters $^{4}_{8}$ in the second position of the **Sync Pattern** field. Figure 11-7 shows the full pattern−sync entry.

$^{4}_{8}$ will be the first character transmitted in the half−duplex 2047−bit test. Since some algorithms are based on strings longer than eight bits (see Figure 11-2), the preceding character ($^{8}_{1}$) was included in the **Sync Pattern** field; but this character is not transmitted.

When it is receiving and analyzing, the test synchronizes on the character in the second position in the **Sync Pattern** field ($^{4}_{8}$ in our example). If there is an error in one of these first eight bits, synchronization will be missed and the test will record a "fault." See Number Of Faults, Section 11.9(G). Following the sync character, erroneous bits are recorded as BIT ERRORS.

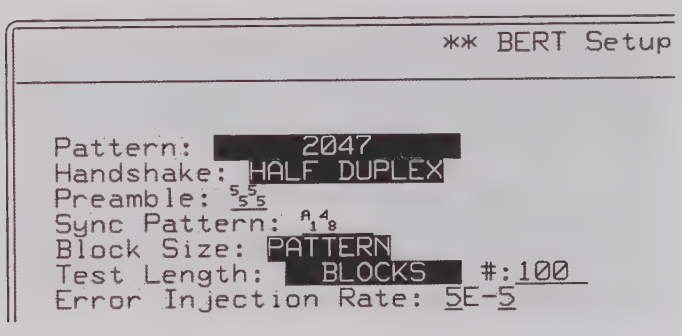


Figure 11-7 Pseudorandom tests in half duplex require pattern–syncing.

In certain cases, two bytes of sync may be insufficient to begin a half–duplex pseudorandom test. The 32767 pattern, for example, requires 15 bits to calculate the 16th bit and succeeding bits. If **Bits:** 8 is selected on the Line Setup screen, a two–byte entry in the **Sync Pattern** field is sufficient. But if a smaller bit–number is selected, two hex characters in the **Sync Pattern** field will not represent the number of bits that the algorithm requires to continue.

In such cases, use the four places in the **Sync Pattern** field (in conjunction with the bit number in the Line Setup) to construct a pattern that is at least as long as the algorithmic series (see Figure 11-2). The last character in the pattern will be the first byte transmitted and analyzed.

3. *Default pattern–sync*. There are two default sets of half–duplex sync points in the INTERVIEW. These are outlined in Table 11-3. BERT testers that use set #1 include the TREND *Data Tester 100* series. NAVTEL's *Datatest II Plus* uses default set #2.

   To operate with sync–point set #1, enter 5 5 in the **Sync Pattern** field. To operate with set #2, clear the field (press CLEAR). If fewer than two characters are entered in the **Sync Pattern** field, the unit will default to set #2. Remember that the default sets are enabled only when **Handshake:** HALF DUPLEX is selected.

Table 11-3
Half–Duplex BERT: Default Synchronization Points

Pattern	1st 8 Data Bits Transmitted	Preceding 8 Bits in Pattern	Entry in Sync Chars Field
Set #1			
63	$^{A}_{3}$	$^{0}_{8}$	$^{S}_{Y}$ $^{S}_{Y}$ or $^{0}_{8}$ $^{A}_{3}$
511	$^{9}_{6}$	$^{A}_{1}$	$^{S}_{Y}$ $^{S}_{Y}$ or $^{A}_{1}$ $^{9}_{6}$
2047	$^{9}_{6}$	$^{B}_{D}$	$^{S}_{Y}$ $^{S}_{Y}$ or $^{B}_{D}$ $^{9}_{6}$
4095	$^{5}_{7}$	$^{A}_{1}$	$^{S}_{Y}$ $^{S}_{Y}$ or $^{A}_{1}$ $^{5}_{7}$
32767	$^{9}_{6}$	$^{F}_{F}$	$^{S}_{Y}$ $^{S}_{Y}$ or $^{F}_{F}$ $^{9}_{6}$
Set #2			
63	$^{B}_{C}$	$^{A}_{3}$	no chars or $^{A}_{3}$ $^{B}_{C}$
511	$^{9}_{6}$	$^{A}_{1}$	no chars or $^{A}_{1}$ $^{9}_{6}$
2047	$^{4}_{8}$	$^{A}_{1}$	no chars or $^{A}_{1}$ $^{4}_{8}$
4095	$^{5}_{7}$	$^{A}_{1}$	no chars or $^{A}_{1}$ $^{5}_{7}$
32767	$^{9}_{6}$	$^{F}_{F}$	no chars or $^{F}_{F}$ $^{9}_{6}$

(F) Block Size

A block can be the length of a cycle or message (PATTERN), or 1000 or 10000 bits rounded to the nearest byte. A block is a component of a test: complete tests often are measured in blocks.

> *CAUTION: The definition of a block varies from standard to standard and from BERT tester to BERT tester. Some standards define a block as the pattern length while others specify 1,000 bits. The user must ascertain and then select the proper definition.*

When a FOX MESSAGE or MESSAGE BUFFER pattern is chosen, PATTERN means one fox message or one message buffer. The PATTERN selection is not valid for the ALT 1010 pattern: a block in this pattern always is a set number of bits.

When a pseudorandom pattern is chosen, the message is one cycle of the pattern. The shortest pseudorandom block is 63 bits, while the longest block is 32,767 bits.

In half–duplex BERT, each transmission is one block. The line turns around following each block. After turnaround, the test continues.

(G) Test Length

Tests are measured in blocks, bits, or seconds. They can also be CONTINUOUS.

The **Test Length:** BLOCKS selection brings up a four−digit # field that accepts entries from 1 to 9999. The shortest pseudorandom block is 63 bits, so the shortest pseudorandom test that is one block long is also 63 bits. The longest test measured in blocks will be 9999 times the longest pattern (32,767), or 327,637,233 bits.

The **Test Length:** SECONDS field also brings with it a # field, with five places for numbers from 1 to 99,999. The maximum number of seconds comes to slightly more than twenty−seven and three−quarter hours.

If **Test Length:** BITS is selected, a # field appears with these rotating selections: 1000, 10000, 100000, and PATTERN.

(H) Automatic Error Injection

Errors can be injected in a bit pattern automatically at a preselected rate. (They may also be injected manually from the keyboard: see Section 11.7 below.) The **Error Injection Rate** field defaults to 5E−5, equivalent to 5 errors per 100,000 bits. The two fives are variables in this formula. The first five is the error rate and can be overwritten with numbers in the range of 0−9. The second five is the **negative exponent** and can be changed to any number in the range of 2−9.

Read the "E−" in the formula as *per 10−to−the−exponent bits*. For example, 1E−2 means $1/10^2$ or 1 error per 100 bits. 0E−6 means $0/10^6$ or no errors per 1,000,000 bits, equivalent to zero. Using 0 as the first variable is equivalent to injecting *no* errors and, in effect, disabling the field.

The highest automatic−error rate selectable would be 9E−2 or $9/10^2$ or 9 errored bits per 100 bits. The lowest rate of injected errors would be 1E−9 or $1/10^9$ or 1 error per 1 billion bits.

11.6 Transmission Format: Line Setup Menu

Certain selections on the Line Setup menu will affect the pattern of bits transmitted during a BERT test. The screen is illustrated in Figure 11-4.

(A) Code

If your BERT pattern is a fox message or a user−defined message that contains alphanumerics, the **Code** that you select on the Line Setup menu will affect the pattern of bits in your test. If your test involves sync characters, remember that the bit pattern for SYN SYN is different for ASCII and EBCDIC.

Testers on either side of a transmission link should be configured for the same code.

(B) Bits

Select the number of information bits. This field is invalid for pseudorandom and alternating 1/0 BERT patterns in SYNC format. In all other BERT configurations, characters are formed according to the bit−number specified here.

(C) Parity

For 5, 6, or 7 information bits in async or isoc format, you may select the type of parity. (For 8 information bits, even, odd, or no parity is available in async or isoc format. Eight bits plus mark or space parity is not available in any format.) The parity bit is additional to the information bits.

The BERT test transmits and checks parity. It calculates parity on the data bits it *expects* to receive, not the actual data bits. This is to prevent an errored data bit from causing a parity error and being counted twice as a result.

(D) Format

Choose **Format:** SYNC , BOP , ASYNC , or ISOC .

1. *Sync.* If **Format:** SYNC is selected on the Line Setup menu and a pseudorandom (or alternating 1/0) pattern is the BERT Setup selection, the pattern will be transmitted bit for bit without synchronization or character−framing. The Line Setup selection fields from **Code** down to (but not including) **Bit Order/Polarity** are invalid.

   If a character−oriented (fox or message−buffer) pattern is selected on the BERT Setup screen, the precise bit pattern will be determined partly by the **Code**, **Bits**, and **Parity** selections on Line Setup. The fields from **Format:** SYNC down to **Bit Order/Polarity** are invalid for BERT, however. Sync characters are entered on the BERT Setup screen. See Section 11.5(E).

2. *BOP.* This softkey is nonfunctional. The selections default to the same as if **Format:** SYNC were selected.

3. *Async.* If **Format:** ASYNC is selected on the Line Setup menu, stop bits (ones) and a start bit (zero) will be added after every fifth, sixth, seventh, eighth, or ninth bit in the BERT pattern, depending on the **Bits** and **Parity** selections on Line Setup. This start/stop−bit framing applies to pseudorandom patterns as well as to character−oriented patterns.

   Do not run the alternating 1/0 pattern if you have selected asynchronous (or isochronous) start/stop−bit framing on Line Setup.

4. *Isoc.* This is a cross between async and sync. It uses asynchronous start/stop−bit framing; but unlike async, internal clock (if selected) will transmit clock pulses on the clock lead(s) for use by the other device on the interface.

(E) Clock

If clock is to be supplied by a modem during the test, you can select **Clock:** EXTERNAL. If no external clock is to be supplied, select INTERNAL or INTERNAL SPLIT and the correct speed or speeds.

> **Note on synchronous and BOP operation:** The INTERVIEW can transmit using internal clock (SCTE) when it is emulating a DTE; but in order to display its own transmit data as well as receive data, it must be connected to a modem or other DCE device that is providing SCT and SCR clock.
>
> If you are have selected **Mode:** BERT DTE and external clock is not available for synchronous or BOP operation, choose **Clock Source:** INTERNAL and patch SCTE to SCT.

(F) Bit Order/Polarity

The pattern of bits in sync and async tests will be affected by the selection in the **Bit Order/Polarity** field only if a character–oriented (fox or message–buffer) pattern is selected on the BERT Setup screen.

Table 11-4
Sync or Async BERT (Line Setup screen)

FORMAT:	SYNC	ASYNC	ISOC
Mode:	BERT DTE or BERT DCE	BERT DTE or BERT DCE	BERT DTE or BERT DCE
Code:	EBCDIC, ASCII, EBCD, XS−3, IPARS, SELECTRIC	EBCDIC, ASCII, EBCD, XS−3, IPARS, SELECTRIC	EBCDIC, ASCII, EBCD, XS−3, IPARS, SELECTRIC
Bits:	8, 7, 6, or 5	8, 7, 6, or 5	8, 7, 6, or 5
Parity:	none	none, even, or odd; mark or space (except 8 bits)	none, even, or odd; mark or space (except 8 bits)
Stop bits:	N/A	1 or 2	1 or 2
Clock source:	external, internal, or internal split	internal or internal split	external, internal, or internal split
Speed:	selectable (except external)	selectable	selectable (except external)
Bit order/polarity:	normal, rev−nor, nor−inv, rev−inv	normal, rev−nor, nor−inv, rev−inv	normal, rev−nor, nor−inv, rev−inv

11.7 Run Mode: Keyboard Control

Whenever you press [RUN], the INTERVIEW will begin to operate as both a BERT generator and analyzer.

It will begin immediately to transmit the bit pattern chosen on the BERT Setup menu according to the format entered on the Line Setup menu. The pattern will repeat until the test ends: see Test Length, Section 11.5(G).

(A) Freezing the Test

The pattern can be interrupted from the keyboard. Pressing [PROGRAM] or [FREEZE] will drive the analyzer out of sync and stop the counters. Freeze mode retains the latest results display on the screen (Figure 11-8). Hitting [FREEZE] a second time will unfreeze the analyzer and resume the count from the frozen readings.

(B) Restarting the Test

The [FREEZE] key restarts a frozen test.

To restart a test while it is running, use the F4 softkey, labeled RESTART. This restart also reinitializes the syncing process. To clear and restart the counters, press F3, RESET. Softkey selections in Run mode are illustrated in Figure 11-8.

Hitting PROGRAM and RUN will also restart everything—test, syncing process, and counters.

(C) Manual Errors

You may introduce errors into the BERT transmission one at a time via softkey. One errored bit will be sent each time the operator presses F1 (INJ1ERR).

(D) Automatic Error Injection

Automatic error–injection can be turned on and off by softkey. Press F2, ERR INJ, to toggle this function. See Section 11.5(H), for an explanation of error rates.

(E) Clearing the Results Screen

To clear the counters without losing sync, press F3, RESET.

(F) Restarting the Test Function

The test function length was determined on the BERT Setup screen to be measured in blocks, bits, or seconds, or to run continuously. Pressing F4, RESTART, restarts the test and reinitializes the syncing process.

(G) Disabling Transmission

You may prevent the BERT pattern from being transmitted while the INTERVIEW analyzes a received pattern. If the unit is in BERT DTE mode, move the breakout switch for pin 2 on the test–interface module (TIM) to the open position. If the unit is in BERT DCE mode, open pin 3 instead.

11.8 Run Mode: Status Line

The status line of the Run–mode BERT display, shown in Figure 11-8, identifies the BERT test and the parameters chosen on the Line Setup menu. The INTERVIEW is using these parameters both to transmit and to analyze. Figure 11-8 shows a BERT 511 test in EBCDIC code; with 8 information bits, no parity, and full duplex pattern; and in synchronous mode.

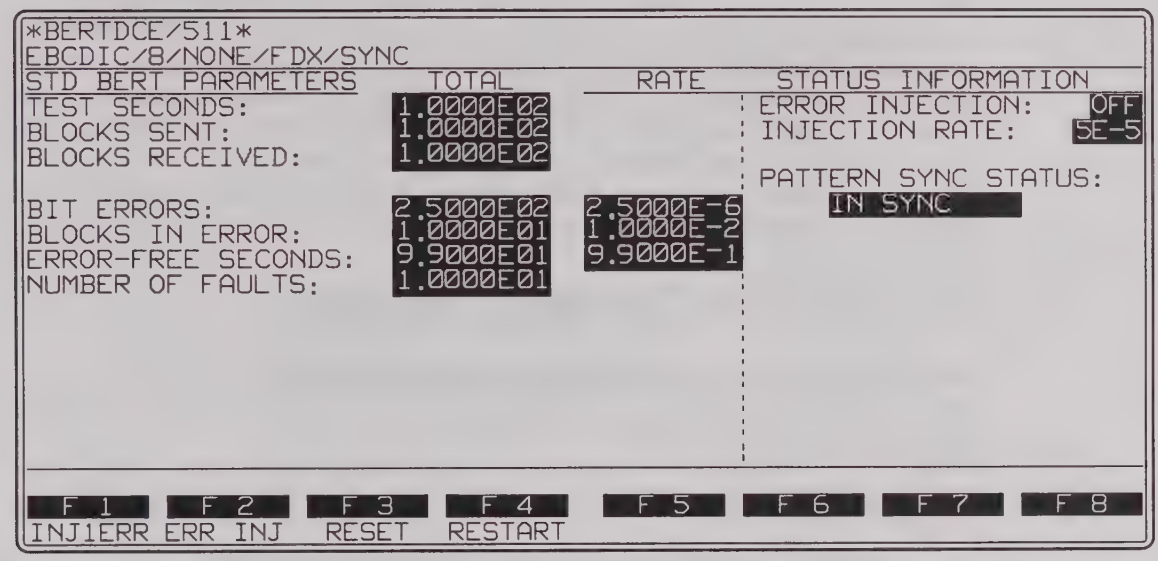


Figure 11-8 The BERT Results screen displays setup status, receiver–sync status, error injection rate, seven counters, and three rate calculations.

11.9 Run Mode: Statistical Display

BERT results are displayed on the Run–mode BERT Statistics screen. You may access this display by pressing the STATS softkey on the top–level softkey rack, followed by the BERT softkey.

BERT counters increment to $2^{64}-1$, or approximately 1.8 times 10^{19}. For practical purposes these counters are unlimited.

Each counter enters Run mode at 0.0000E00. The top of Figure 11-9 shows the block counter on the verge of rolling over to a new exponential value. The bottom of the figure shows the effect of the next block received: the exponent has incremented so that the counter is being updated on every tenth count instead of every one. The counter will be updated next when ten new blocks have been received.

BLOCKS RECEIVED: 9.9999E04

BLOCKS RECEIVED: 1.0000E05

Figure 11-9 This is a before–and–after illustration of the block counter receiving its 100,000th block.

Three of the counters have Rate displays adjacent to them. See Figure 11-8. These rates are to be read in the same way you would read an entry in the **Injection Rate** field on the BERT Setup menu (see Section 11.5(H)): 9.0000E–3 means 9 times $1/10^3$ or 9 errors

per 1,000 bits. Rates are displayed in real time. They enter Run mode at 0.0000E–0 and remain at zero until a bit error or block error or errored second occurs. Once an error rate is posted, the displays will behave like decrementing counters: while the number of bits, blocks, or seconds increases steadily, the rate of error will decrease. The top of Figure 11-10 shows a Rate display for bit errors that stands at 1.0000E–4 or 1 error per 10,000 bits. When the next bit arrives, unless it is an error the rate will decrease. The bottom of the figure shows the action of the display as the rate of error decreased: 1 per 10,000 became 9.9999 per 100,000 (9.9999E–05).

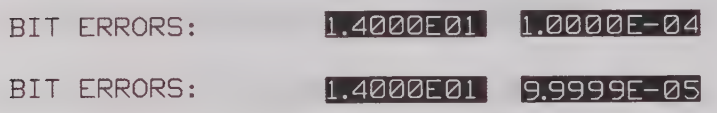


Figure 11-10 As the rate of error decreases, the minus–exponent grows larger.

(A) Test Seconds

This counter is incremented once for every second the test runs.

(B) Blocks Sent

The current number of blocks sent is recorded here. A block is defined in Section 11.5(F).

(C) Blocks Received

This shows the current number of blocks received. If this count is not incrementing, the INTERVIEW is not syncing on a pattern and the **PATTERN SYNC STATUS** field on the lower right of the BERT statistical display screen should say OUT OF SYNC.

(D) Bit Errors

The bit sequence received is compared with that expected in accordance with the parameters chosen on the Line Setup menu and the BERT Setup screen. The count of received bits that do not match the expected pattern is displayed here. This counter value in relation to total bits is given in the Rate column at the right of the counter, expressed in errors per exponent of 10.

(E) Blocks In Error

The number of blocks in which one or more bit errors have occurred is recorded. A rate calculation of this value to total blocks received is given to the right of the counter in real time.

> **NOTE:** If two testers are being used, verify that both are defining block size in the same way. See Section 11.5(F).

(F) Error–Free Seconds

This counter will increment when the Test Seconds counter increments until one or more errors have been found during the last second. The rate of this counter value compared to total seconds is given to the right of the counter in real time.

(G) Number Of Faults

"Fault" has different meanings for full and half duplex. In full duplex, a fault is recorded whenever an error is detected in more than 25 percent of the bits received over a certain period of time (approximately 16 bits in 64 contiguous bits). This percentage is considered sufficient to indicate that a bit time has been missed.

In full duplex, if **Resync:** ON has been selected on the Transmit Mode menu the INTERVIEW's receiver will resynchronize every time a fault is found.

In half duplex, a fault occurs when the analyzer sees bit transitions that indicate a new block of BERT data but fails to recognize the sync pattern or the sync character.

(H) Error Injection

The status of Error Injection may be ON or OFF; it is toggled manually by the ERR INJ softkey. In Run mode it is turned on and off by the F2 softkey.

(I) Injection Rate

This status field simply reports the error–injection rate that the user has entered on the BERT Setup menu.

(J) Receiver

BERT analysis begins when the receiver synchronizes on incoming data. The **PATTERN SYNC STATUS** line on the results screen reports on receiver status. At all times during a BERT run, the line displays one of two messages, IN SYNC or OUT OF SYNC.

Once synchronization is established, the receiver can go out of sync only if **Resync:** ON is selected. During the out–of–sync condition, data is not analyzed for error.

When the resync function is turned off, the IN SYNC condition, once established, will remain in effect until the test ends. This setup is appropriate for relatively brief tests on noisy circuits.

11.10 Loopback at the Transmitting INTERVIEW

The INTERVIEW can analyze its own BERT transmission without being connected to the data interface. This is a good way for you to become familiar with the BERT test procedure before you apply it to a system.

With the interface disconnected, power up the unit and select **Mode:** BERT DCE on the Line Setup menu. Then select a configuration compatible with the specifications in Table 11-2 and Table 11-3. Select internal clock.

Press [RUN]. You will see only the first two counters, Test Seconds and Blocks Sent, incrementing. This is the way the Run−mode display will appear when you are sending the BERT pattern to another BERT analyzer. No statistical analysis is being done because the BERT analyzer is not seeing the transmitted pattern. The report on Receiver status is OUT OF SYNC.

Press [PROGRAM]. On the breakout box on the test−interface module (TIM), patch TD to RD. Press [RUN]. This time you will see the first three counters incrementing: Test Seconds, Blocks Sent, and Blocks Received. The receiver is now IN SYNC. The INTERVIEW's BERT analyzer can see its own transmission but it is unlikely that it will find any errors in its own data.

Use the [F1] key to introduce errors into the transmission. Observe the next three counters and the rate measurements alongside them. The [F1] key will not introduce a fault.

Run another test into which you have injected automatic errors. An **Error Injection Rate** entry of 1E−5 on the BERT Setup screen will produce one errored bit for every 100,000 bits and a bit−error rate of 1.0000E−5.

12 Standard Interfaces

The INTERVIEW contains a universal logic interface that supports data rates into the megabits–per–second range. Physical interfaces that are adapted to the various data rates are provided in the form of test–interface modules (or TIMs).

The INTERVIEW 8000 Series offers you a choice of standard TIMs (and, for some units, MUX board selections), depending on the configuration chosen. The TIM you select will have its own specific documentation with it. For optional TIMs currently available, see Section 13. Some interface module provides *breakout* patching and switching for each lead.

TIMs are modular and simple to install. There are two steps to installing an interface module: 1) **With the unit powered off,** insert the TIM into the module slot at the rear of the unit and press until it latches; 2) apply the proper LED overlay in position on the front of the unit above the screen. The overlay is a flexible plastic strip with a small tip on each end that fit into notches in the front panel. The overlay covers the front–panel green–red LEDs and gives them connector–specific identification.

The test–interface module locks into place in the back of the unit and a small release bar must be pressed to unlock it. The rear of the unit is illustrated in Section 1 of this manual. The interface softkey (F2 on the Setup Menu) will reflect the type of TIM installed, such as RS232, V.35, RS449, RC8245, TTL, etc.

13 Optional Interfaces

In addition to the standard TIMs available, there are several optional TIMs available for purchase. Table 13-1 lists currently available TIMs and their option numbers. Telenex offers a TIM expansion shelf (OPT−951−42−1), which connects up to five TIMs for easy use—no changing TIMs when changing single−port interfaces; just change your connections to the line. Note that dual−port and ISDN TIMs are not compatible with INTERVIEW 8100 units. For further information, contact Customer Service.

Table 13-1
Optional Test Interface Modules

TIM	Option Number
V.35 (North America)	OPT−951−510−1
V.35 (Metric)	OPT−951−510−2
T1/ISDN PRI	OPT−951−511−2
RS−449	OPT−951−512−1
X.21	OPT−951−513−1
RS−232	OPT−951−514−1
RC−8245 (RS−485, NRZI)	OPT−951−521−2
G.703/ISDN PRI	OPT−951−524−1
TTL	OPT−951−545−1
Dual−Port G.703/ISDN PRI (75 ohm)	OPT−951−555−1 (TIM with MUX) OPT−951−564−1 (TIM only)
Dual−Port G.703/ISDN PRI (120 ohm)	OPT−951−555−2 (TIM with MUX) OPT−951−565−1 (TIM only)
Dual−Port T1/ISDN PRI	OPT−951−555−3 (TIM with MUX) OPT−951−627−1 (TIM only)
Dual−Port G.703/ISDN PRI (75 ohm) plus ISDN Basic Rate Interface	OPT−951−556−1 (TIMs with MUXes)
Dual−Port G.703/ISDN PRI (120 ohm) plus ISDN Basic Rate Interface	OPT−951−556−2 (TIMs with MUXes)
Dual−Port T1/ISDN PRI plus ISDN Basic Rate Interface	OPT−951−556−3 (TIMs with MUXes)
ISDN S/T/U (TIM, MUX, & Handset)	OPT−951−563−1
ISDN S/T/U (BRI Add−on TIM & MUX)	OPT−951−566−1
Dual−Port DS−0A	OPT−951−569−1 (TIM with MUX) OPT−951−571−1 (TIM only)
Dual−Port G.7603/64KBPS Co−Directional	OPT−951−570−1 (TIM with MUX) OPT−951−572−1 (TIM only)
Dual−Port V.35/EIA−530(449)/EIA−232	OPT−951−620−1 (TIM with MUX) OPT−951−620−2 (TIM with MUX) OPT−951−621−1 (TIM only) OPT−951−621−2 (TIM only)

14 Disk Maintenance

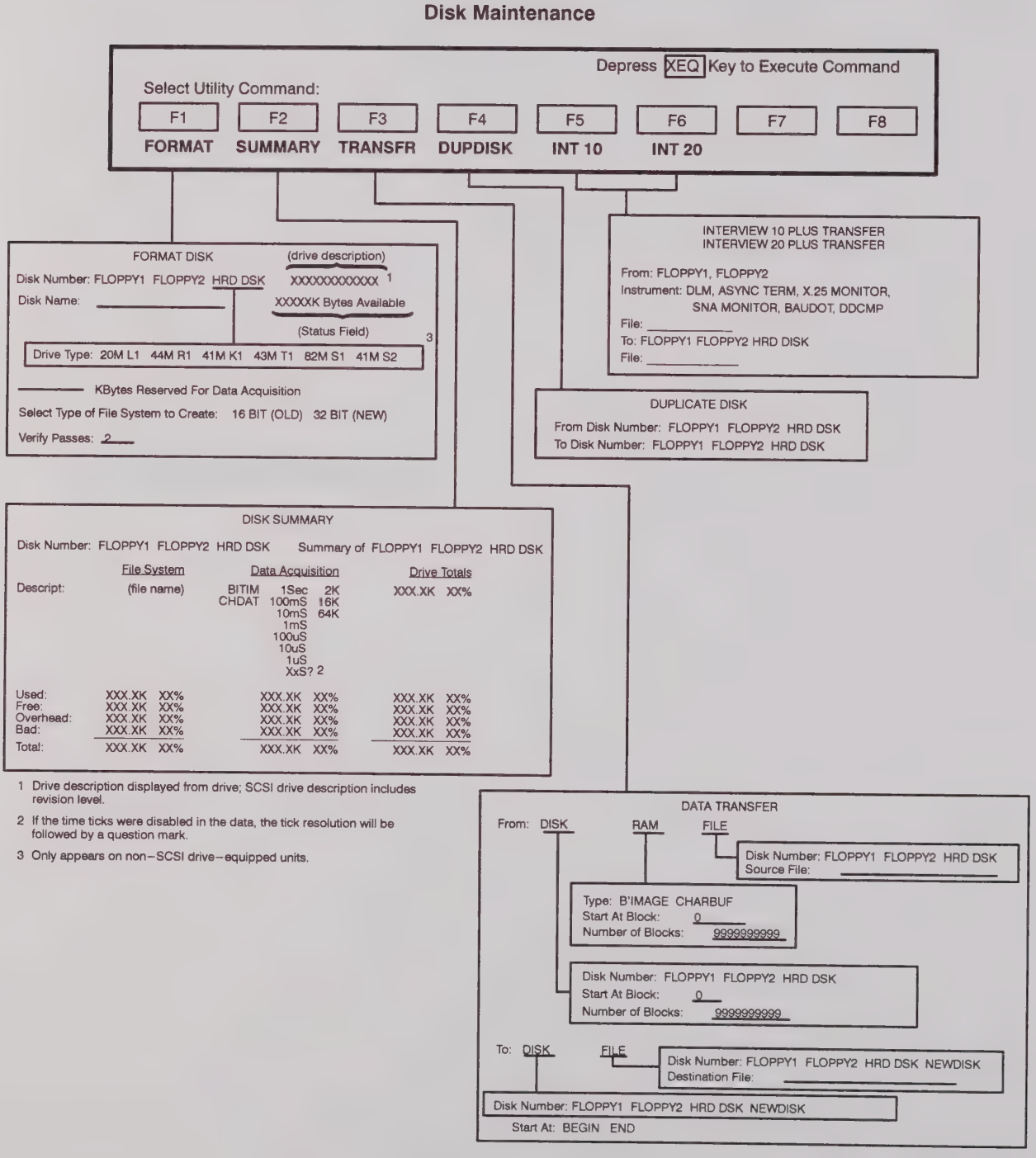


Figure 14-1 Disk Maintenance menu.

14 Disk Maintenance

14.1 The Disks

The INTERVIEW 8200, 8600, 8700, and 8800 *TURBO* units have two disk drives designed for 3.5 inch double–sided, high–density microfloppy disks; the INTERVIEW 8100 and the INTERVIEW 8000 Series remote units have one such disk drive. Maximum storage capacity of each microfloppy is 1.4 Mbytes.

The maximum storage capacity of the hard disk drive in the INTERVIEW 8000 Series units varies with the installed drive. (Note that hard disk availability within the industry may increase disk capacity in future units as necessary.)

The drives are referenced by number. Floppy Disk 1 (FD1) is the disk installed in the left–hand drive. Floppy Disk 2 (FD2) resides in the right–hand drive. The Hard Disk drive is referred to as HRD.

14.2 Allocating Disk Space

Each disk must be formatted before it is available for data capture. Memory on each INTERVIEW disk can be partitioned for two types of storage: data acquisition tracks and a filing system. This partitioning must be performed before information is stored on disk. The Format command on the Disk Maintenance screen is used for this purpose. This command is explained later in this section.

14.3 Data Acquisition Tracks vs. the Filing System

Data acquisition tracks are sequential access data tracks which store Bit–Image or Character–oriented data in block format. Minimum block size is 2 Kbytes, and each recorded block is numbered and date/time stamped. When data is recorded on disk in real–time, it can be captured on data acquisition tracks or to a file.

Recorded blocks can be accessed by block number and played back. However, any subsequent disk recording session to DAT overwrites the contents of the data acquisition tracks, so it is advisable to save this recorded data to a file using the Data Transfer command on the Disk Maintenance screen (see Section 14.4). Any subsequent disk recording session to a previously recorded file will also overwrite the file.

The filing system is a hierarchy of user–created files and directories for storing and organizing captured line data, setups (menu contents), protocol package data, C code, or

entire test programs. The user can format the hard disk drive to create a 32–bit file system in lieu of the standard 16–bit file system. Files are identified and accessed by name (full or relative pathname). Existing files and directories, descriptive information, and file management commands all appear on the File Maintenance screen. The File Maintenance screen, file naming conventions, and file access are discussed in Section 15.

14.4 The Disk Maintenance Screen

The Disk Maintenance screen is part of the Utilities menu group. All selections available on the menu are shown in Figure 14-1. The default Disk Maintenance screen is shown in Figure 14-2. Several disk management functions are grouped on this menu (as evidenced by the function keys in Figure 14-2). Disk formatting, data transfers, and disk duplication are all performed from this menu. In addition, the menu provides a summary of disk space for each of the disks associated with the INTERVIEW.

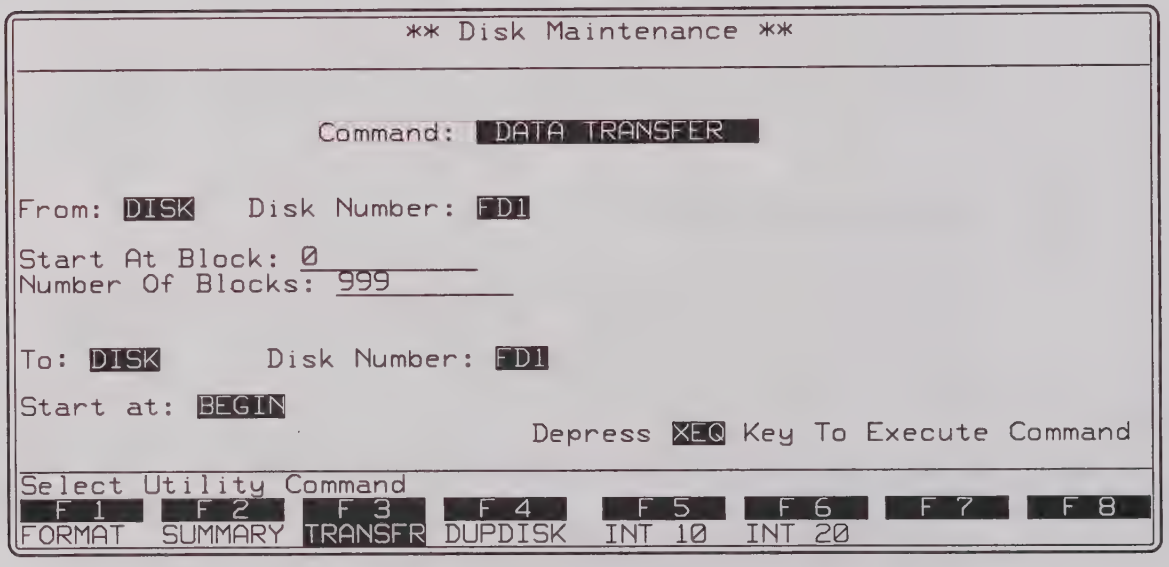


Figure 14-2 The default screen.

The selected command is always posted at the top of the screen. The screen repaints for each command. Once you have filled in appropriate menu fields, press XEQ to execute the command. In cases where data could be destroyed, you will have the option to continue the process by pressing GOAHEAD, or to stop the process by pressing ABORT. Status messages will inform you of any errors that are encountered. Error messages are listed and defined in Appendix A.

(A) Format Disk

This command (see menu selections, Figure 14-3) is used primarily to prepare floppy disks for data acquisition and file creation. A new disk must always be formatted before use.

The hard disk is formatted prior to delivery. *It is not recommended that you format the hard disk.* If you do want to format it to reallocate storage space, however, be sure that you know the size of your hard disk before you begin. Use the Disk Maintenance Summary command to check the size. See Section 14.4(B). The total in the right−hand column should approximate 82 Mbytes for the older−model standard Winchester drive or 240 Mbytes or greater for SCSI drives, depending upon the actual drive installed. You may also wish to reformat your hard disk drive to a 32−bit file system; see subsection 5. on the **File System Type** field below.

> *CAUTION: Formatting a disk causes loss of its entire contents. It is recommended that you write−protect or duplicate any disk you wish to preserve. Periodic back−up of the hard disk is strongly recommended. Backup is* required *prior to formatting the hard disk. See Section 2.4 for instructions on backing up the hard disk.*

Before this command is executed, a number of subfields must be filled in. Subfields appear when you press the FORMAT function key.

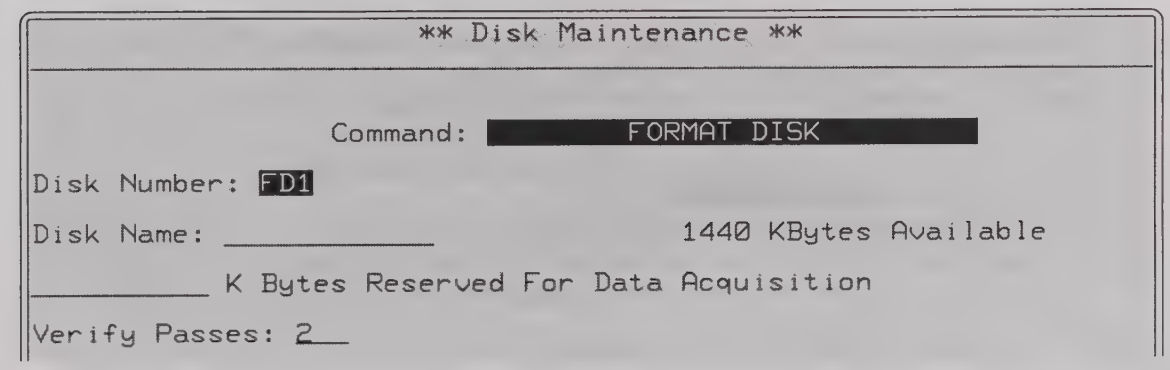


Figure 14-3 New disks must be formatted for data acquisition and file creation.

1. *Disk number.* Floppy 1, Floppy 2, or Hard Disk must be designated for formatting in the **Disk Number** field. Once the disk is selected, a status field appears to the right, indicating the total number of Kbytes of storage available on the disk.

NOTE: Floppy 1 (FD1) refers to the INTERVIEW's left–hand micro–floppy drive; Floppy 2 (FD2) to the right–hand drive. The abbreviation HRD represents the hard disk drive.

2. *Disk name.* When one of the floppy drives is selected, the **Disk Name** field appears. A disk name is optional; however, it provides a useful identification for each disk.

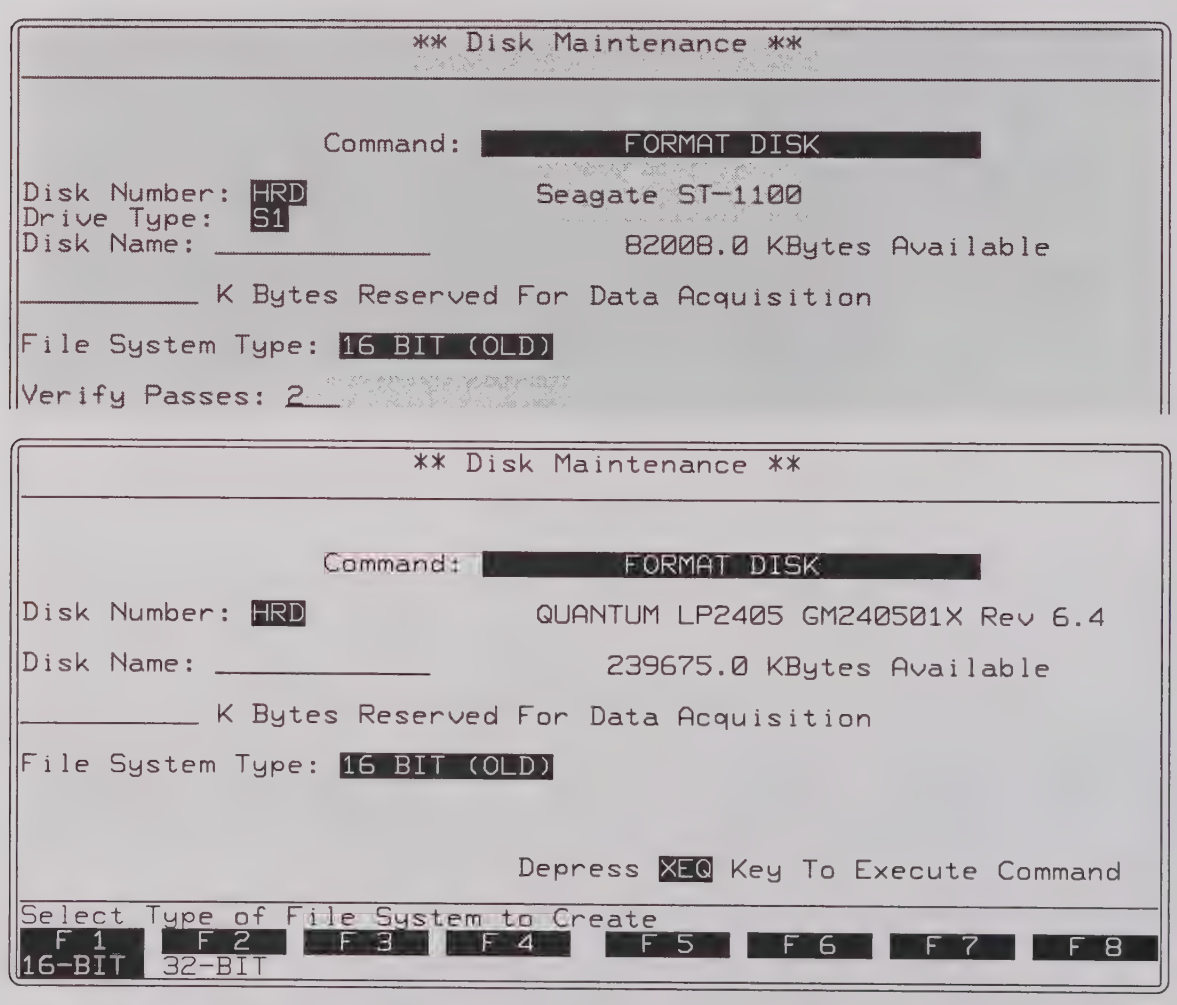


Figure 14-4 Indicate the size of the hard disk installed in your unit on non–SCSI units; SCSI units automatically display the drive information.

3. *Drive type.* When you select **Disk Number:** HRD , the **Drive Type** field appears on ST506 units; SCSI units automatically read the information from the drive itself.

See Figure 14-4. Select from 20M L1, 44M R1, 41M K1, 43M T1, 82M S1, or 41M S2 to indicate the size of your hard disk; the INTERVIEW 8000 Series defaults to 82M S1. If you try to format a disk to a size larger than its capacity, you will get an error message. If you inadvertently format a disk to a size smaller than its capacity, you may reformat it to the correct size.

4. *Allocate space for data acquisition.* The next menu field requests an entry for the amount of space on the disk which is to be dedicated to the data acquisition tracks. Legal entries in this field are 0 to 1322 for micro–floppies. (Higher entries will be accepted, but no additional space will be allocated.) If the field is left blank, no space is allocated for data storage.

   Space allocation on the hard disk depends on the size of hard disk installed in the unit:

   - Seagate 82–Mbyte hard disk
   - SCSI 240–Mbyte hard disk
   - SCSI 270–Mbyte hard disk
   - SCSI 1.2–Gbyte hard disk
   - SCSI 2.1–Gbyte hard disk

   The amount of space allocated to data acquisition tracks is subtracted from the total Kbytes available (as listed in the status field on this menu). When disk space is allocated to data acquisition tracks, the amount of space is always rounded upwards to an integral number of cylinders. Remaining Kbytes are automatically allocated to the filing system. At least two cylinders of disk space (36 Kbytes) are always reserved for the file system on floppy disks. On the hard disk, a minimum of eight cylinders (612 Kbytes) is reserved for the file system.

   The factory standard partition for file storage is 32M when the (old) 16–bit file system is chosen in the **File System Type** field. The factory standard partition for file storage when utilizing the 32–bit file system is 500M.

5. *File System Type.* This field only appears when **Disk Number:** HRD is selected. The original 16–bit file system was limited to a maximum size of 32 Megabytes; the new 32–bit file system is limited only by the size of your hard disk drive. The factory standard partition for the 32–bit file system is 500 Mbytes; there is no maximum.

   The **File System Type** menu field defaults to the old 16–bit file system size (see Figure 14-4). Select 32–bit to format the hard disk drive to the larger file system.

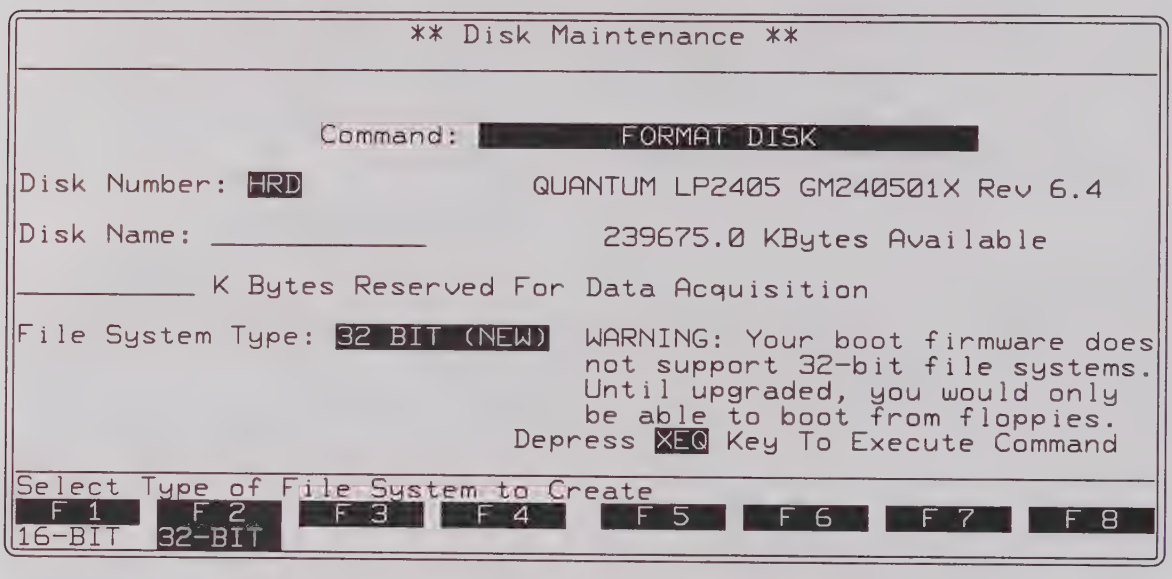


Figure 14-5 The file system type can be formatted in 16–bit or 32–bit format.

If you do not have the appropriate firmware revision installed in your unit, the warning on the screen above will appear. You may elect to continue with the file system size increase, but be aware that you will need to boot the unit from floppy disks after the hard disk is reformatted.

CAUTION: Formatting a disk causes loss of its entire contents. It is recommended that you write–protect or duplicate any disk you wish to preserve. Periodic back–up of the hard disk is strongly recommended. Backup is required *prior to formatting the hard disk. See Section 2.4 for instructions on backing up the hard disk.*

6. *Verify Passes*. The **Verify Passes:** field is applicable to all non–SCSI hard drives and floppy disk drives. If the unit is configured for a SCSI drive, this field disappears as the SCSI drives have their own format/verify procedure.

   This field defines the number of media verification passes the unit makes during formatting. The user may now specify a number from *0* to *99*; the default is *2* passes. It is well to note, however, that the higher the number of verifications, the longer the process will take; also, it is *not* recommended that *0* be entered, as the process will then be totally unverified.

Press [XEQ] to start formatting. If the disk selected contains data, the warning message "Formatting disk will destroy data" appears at the top of the screen. *Be sure that you have backed up your data before continuing.* Press GOAHEAD ([F1]) to continue the formatting process. Press ABORT ([F8]) if you decide not to format. After you press [XEQ], you may not abort formatting since the disk has already been altered.

(B) Disk Summary

Press SUMMARY, select the disk you want summarized, and press [XEQ] for a synopsis of disk contents. The **Summary of** field (see Figure 14-6) is updated when you press [XEQ], so that the number of the disk drive being summarized appears in this field.

Several lines of disk information follow with one column for the file system, one for the data acquisition tracks, and the third for the totals for the drive selected. The fields in the Drive Totals column are the sum of the fields of the first two columns. See paragraph 6. below for the summation information of the **Total:** fields.

```
                    ** Disk Maintenance **

                    Command:  DISK SUMMARY

Disk Number: FD1                Summary of FD1

                File System      Data Acquisition      Drive Totals
Descript:     Sample #001          BITIM 1mS 2K
Used:          5.5K  15%          1324.0K  94%        1329.5K  92%
Free:         11.0K  30%             1.5K   0%          12.5K   1%
Overhead:     19.5K  54%            78.5K   6%          98.0K   7%
Bad:           0.0K   0%             0.0K   0%           0.0K   0%
Total:        36.0K   2%          1404.0K  97%        1440.0K 100%

                         Depress XEQ Key To Execute Command

Select Utility Command
 F 1     F 2     F 3     F 4     F 5     F 6     F 7     F 8
FORMAT  SUMMARY TRANSFR DUPDISK INT 10  INT 20
```

Figure 14-6 Results of a *DISK SUMMARY* command.

1. *Description.* There are two types of **Descript:** fields. The File System column field gives the name of the disk as it was entered in the **Disk Name** field when the disk was formatted.

   The Data Acquisition column field indicates the type of data (**BITIM** or **CHDAT**), the rate at which it was recorded (**1Sec, 100mS, 10mS, 1mS, 100uS, 10uS,**or **1uS**) or whether TD bits or RD bits were selected (**TDclk** or **RDclk**), and the DAT record size (**2K, 16K,** or **64K**).

   NOTE: If the time ticks were disabled in the data, the tick resolution will be followed by a question mark.

2. *Used.* These fields represent the amount of disk space actually used in Kbytes in and as a percentage of the disk space allotted to each of the column headers: the file system, the data acquisition tracks, and the total of the selected drive.

In the Drive Totals columns, the amounts of file space remaining are reported. Note that the total in the right column always equals 100%, while the total in the left column always is 1440 Kbytes for a floppy diskette. For the hard disk, the total should be the size of the drive—82 Mbytes, 240 Mbytes, 2.1 Gbytes, etc. (If you formatted to a smaller size selection, the total would be the size of your selection.)

3. *Free.* These fields represent the amount of disk space "free" or unused in Kbytes in and as a percentage of the disk space allotted to each of the column headers: the file system, the data acquisition tracks, and the total of the selected drive.

4. *Overhead.* The **Overhead:** fields indicate storage that is unusable for holding data; it may be used by the system, for example, as holding critical bookkeeping information. As with the other fields, these fields are expressed in Kbytes and as a percentage of the disk space allotted to each of the columns: the file system, the data acquisition tracks, and the total of the selected drive.

5. *Bad.* These fields represent the amount of disk space storage that was found to be flawed during one of the verify passes when the drive was last formatted; this is unusable space. As with the other fields, these fields are expressed in Kbytes and as a percentage of the disk space allotted to each of the columns: the file system, the data acquisition tracks, and the total of the selected drive.

6. *Total.* These totals are counts of all blocks in the partition and their respective percentages of the entire drive space. Note that while the total of the percentages above the **Total:** fields line add up to (or approximately to) 100%, the percentages expressed in the **Total:** fields are not 100% as they are fractions of the entire drive space used by the File System and by DAT.

(C) Data Transfer

Use this command (see Figure 14-7) to move data from one storage medium to another. Available media are the following: data–acquisition memory (whether character or bit–image) on a disk; files on a disk; bit–image RAM ("RAM" on the Record Setup menu); or character RAM (the screen buffer). Data transfer is necessary in some cases. For example, data in bit–image RAM, data in the screen buffer, and data saved to a file must be transferred to disk before it may be played back.

> **NOTE:** Transferring data to disk blocks which contain data or to an existing file will overwrite the previous contents of the disk or file.

Press TRANSFR to display subfields for the command.

1. *From RAM, disk, or file.* The **From** field allows you to specify the source of data to be transferred. When you select RAM, DISK, or FILE, the screen repaints. New menu fields are explained at the end of this section.

2. *RAM*. When RAM is selected, you must specify type of RAM and the number and location of blocks you want to transfer. See below, subsection 6.

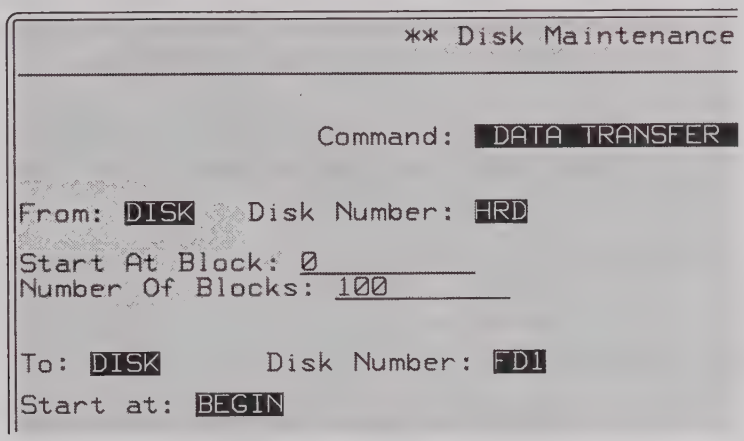


Figure 14-7 The source of transferred data may be RAM, disk, or file; the destination may be disk or file.

3. *Disk*. This selection refers to the data acquisition tracks on disk. When DISK is selected, you must specify the disk number and the number and location of blocks you want to transfer.

4. *File*. This selection allows you to transfer data out of the filing system. When FILE is selected, you must specify the name of the file and the disk number on which it resides. Data in a file is available for later playback.

5. *Disk number*. Designate the source disk by selecting FD1 (left−hand drive), FD2 (right−hand drive), or HRD.

6. *Type*. The first subfield which appears when RAM is selected is **Type**. You must specify BIT IMAGE or CHARBUFFER in the **Type** field when transferring data from RAM.

   Bit−image RAM is the RAM that is selected in the **Capture Memory** field on the Record Setup menu. For "normal" bit−image record, this RAM is 256 Kbyte in both the 8100 *TURBO* and 8200 *TURBO* and 1 Mbyte in the 8600 *TURBO,* 8700 *TURBO,* and 8800 *TURBO*; for "high−speed" bit−image record, this RAM is 256 Kbyte in both the 8100 *TURBO* and 8200 *TURBO*, 1 Mbyte in the 8600 *TURBO*, 2 Mbytes in the 8700 *TURBO,* and 3 Mbytes in the 8800 *TURBO*.

   Character RAM is the 64 Kbyte screen buffer (or "character buffer").

7. *Start at block*. Enter the number of the first block to be transferred from the data source (whether disk or RAM) in the **Start At Block** field. (You can observe the

current block number at the top of the display when you are recording data in Run mode.) Valid entries in the **Start At Block** field are 0 to 9999999999. If you do not enter a value, the value will default to zero. Block 0 means that data transfer will start from the beginning of the data source, regardless of the actual block number. All other entries represent actual block numbers.

> **NOTE:** It is not necessary to enter leading zeroes in this field. For example, you may enter 10 instead of 00010.

8. *Number of blocks.* In this field, enter the total number of blocks you wish to transfer. Valid entries are 1 to 9999999999; however, your entry is bounded by and limited to the size of the destination media. Again, it is not necessary to enter leading zero's.

9. *To disk or file.* When you transfer data to DISK, the data is transferred onto the data acquisition tracks for later playback. Specify the disk by selecting FD1 (left–hand drive), FD2 (right–hand drive), or NEW (for use when you have only one floppy disk drive available and are transferring data from one disk to another.)

   When you transfer data to a FILE, you must indicate the number of the destination disk as well as the file name. (Either relative or full pathname may be given.) Specify the disk by selecting FD1 (left–hand drive), optional FD2 (right–hand drive), or NEW (for use when you have only one floppy disk drive available and are transferring data from one disk to another.) If the name of an existing file is entered, the contents of the file will be overwritten with new data during the transfer. Data in a file is available for later direct playback.

10. *Start at.* Specify in this field whether data will be stored on the destination disk starting at the beginning of the disk or file (overwriting any existing data), or whether the transferred data will be appended to the end of existing data.

Once you press [XEQ], you may stop the transfer at any point by pressing [F8] (ABORT).

(D) Duplicate Disk

If both floppy disk drives are operable, this command allows you to make a copy of full disk contents.

> **NOTE:** Duplicating the contents of one disk onto another destroys the previous contents of the destination disk.

Press DUPDISK to see subfields for this command. In the **From Disk Number** field (see Figure 14-8), select the drive number for the disk to be copied. Select the disk which will receive the copy in the **To Disk Number** field; the field defaults to FD2 .

NOTE: If the disk receiving data is write–protected, the Duplicate command will fail. Otherwise, any data on a receiving floppy disk will be overwritten. Only those files on a receiving hard disk drive *with the same names* as those on the sending disk will be overwritten, but files with new names will be added to those existing on the hard drive.

From time to time, you may need to use the Duplicate Disk command to install new system software on the hard disk of the INTERVIEW (if present). See Section 2.3.

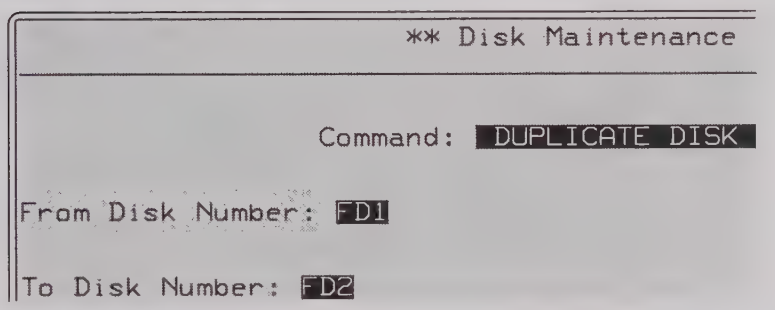


Figure 14-8 Entire disks may be duplicated on the Disk Maintenance menu.

(E) Transferring INTERVIEW 5 Plus, 10 Plus, 15 Plus, or 20 Plus Data to the 8000 Series

The INTERVIEW Transfer command (explained in Section 14.4(F)) allows you to copy files from a disk recorded with an INTERVIEW 5, 10, 15, or 20 Plus onto the hard disk of an INTERVIEW 8000 Series unit, or onto a microfloppy. The INTERVIEW 5/10/15 Plus and 20 Plus Series uses a 3.5 inch double–sided, double–density disk which can be distinguished from the INTERVIEW 8000 Series disk because it has a write–protect/enable window (with a plastic cover) but no second window on the opposite corner.

These disks are compatible (that is, they may be *read* by any 8000 Series unit) if the INTERVIEW 5/10/15 Plus or 20 Plus on which they were recorded has been reconfigured so that it records to disk in the correct format.

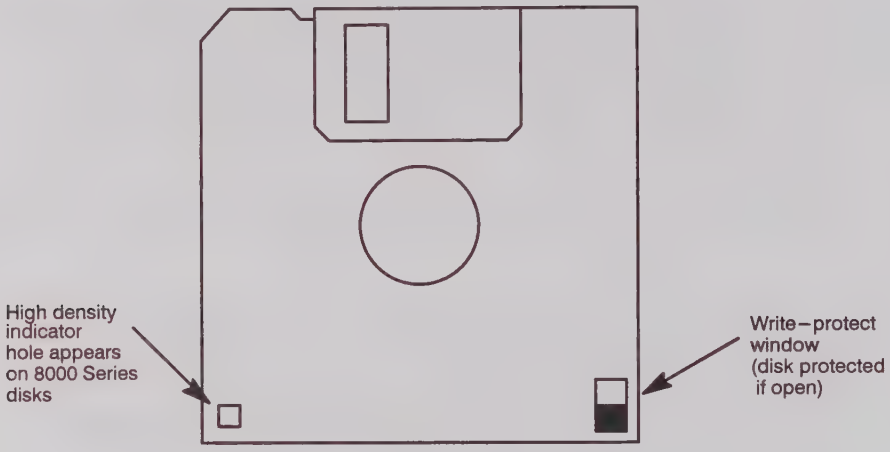


Figure 14-9 The INTERVIEW 8000 Series uses 3.5 inch quad– or high–density, double–sided microfloppy diskettes.

NOTE: We recommend that you record disks reconfigured for the 512 bytes/sector format (Software Version 1.10A or higher). If you are communicating with INTERVIEW 5, 10, or 15 Plus field units which have not yet been reconfigured for this format (still have Software Version 1.00C with old 1024 bytes/sector format), we recommend you contact the factory for instructions on reconfiguring the unit so that data is recorded to disk in the proper format.

Programs for the INTERVIEW 5/10/15 Plus and 20 Plus Series may not be transferred to disk for the INTERVIEW 8000 Series. Furthermore, neither data nor programs recorded on an INTERVIEW 8000 Series unit can be copied or read by the INTERVIEW 5/10/15 Plus or 20 Plus Series. This precludes the storage of data on the 8000 Series for later analysis on the INTERVIEW 5/10/15 Plus or 20 Plus. However, a more sophisticated analysis of this data is possible when it is played back on the INTERVIEW 8000 Series.

All INTERVIEW 5/10/15 Plus files are transferred as ASCII character data, regardless of the format in which the data was originally recorded. Because of the conversion in data format, the size of the file once it is transferred may increase by as much as 5%.

However, all INTERVIEW 20 Plus files are transferred as the same character data format in which the data was originally recorded. (For example, data recorded on an INTERVIEW 20 Plus in EBCDIC format would be transferred as EBCDIC data.)

The resulting data file may be handled as any other character–oriented data. The only difference is that data originally recorded on the INTERVIEW 5/10/15 Plus or 20 Plus Series does not contain the time and date stamp that normally appears at the upper right of the display screens. The time field in a layered protocol trace, likewise, does not contain the actual time that the data was recorded; rather, it contains information on the relative time between frames. The INTERVIEW 8000 Series stamps the first frame in the trace with the time elapsed since [RUN] was pressed. The time displayed for subsequent frames depends on the delay since the start of Run mode. The time between frames will vary for the same data depending on the playback speed.

> **NOTE:** Microfloppies configured for 512 bytes/sector have a maximum storage capacity of 691,200 bytes as opposed to the 766,976 bytes of maximum storage available on the older 1024 bytes/sector disk. Maximum recording speed of the 512 bytes/sector disks is 64 Kbps.

Data samples captured on the INTERVIEW 5/10/15 Plus and 20 Plus Series and saved to disk files may be transferred to the INTERVIEW 8000 Series. You must first, however, know the filenames for the recorded data. If no printout accompanies the INTERVIEW 5/10/15 Plus or 20 Plus disk, you may obtain a listing of disk files using the procedure which follows.

1. Install the INTERVIEW 5/10/15 Plus or 20 Plus disk into the INTERVIEW 5, 10, 15, or 20 Plus.

2. Press the Menu key until you see the start–up screen. This screen displays the name of the device in a banner and gives the copyright date and the software version. The prompt above the function keys instructs you to select an instrument. Press the function key indicated for more instruments.

3. A new set of function key labels appears on the screen. Press the function key indicated for Disk Utilities.

4. *For INTERVIEW 5/10/15 Plus:* The screen repaints to reveal file listings for all files stored on the disk. As shown in Figure 14-10, for the INTERVIEW 5/10/15 Plus, there are four columns labeled at the top of the screen. The first column tells the instrument type for the file named on the same line. The second column tells whether it is a data file (D) or a program file (P). The name of the file appears in the third column. The size of the file is listed in the final column.

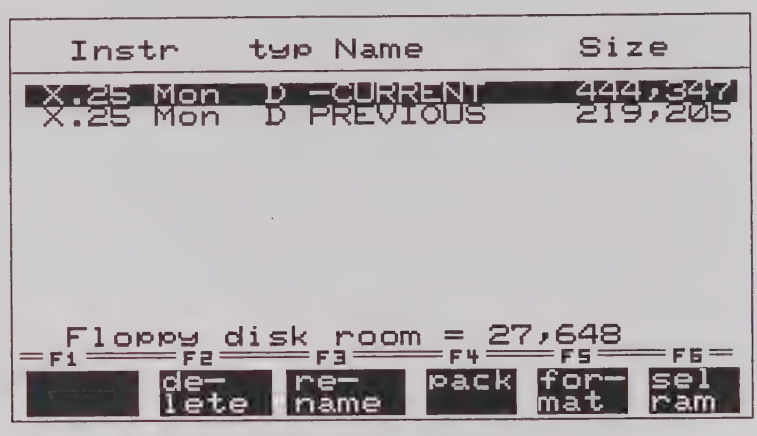


Figure 14-10 INTERVIEW 5/10/15 Plus disk file directory.

NOTE: Data samples which have been saved as a memory buffer rather than a file are not recorded on the microfloppy, will not appear in the file listings, and cannot be transferred to the INTERVIEW 8000 Series.

If the "–CURRENT" file is to be transferred, we recommend renaming it. In the event that another file is saved before the transfer, that new file will overwrite the present "–CURRENT" file. Select the file to be renamed and press the Rename function key; enter a new name for the file as the display directs.

5. *For INTERVIEW 20 Plus:* The screen repaints to reveal file listings for all files stored on the disk. For the INTERVIEW 20 Plus, as shown in Figure 14-11, there are three columns. The first column lists the instrument type for the file named on the same line and the size of the file is listed in the final column. The second column contains three pieces of information:

A:02–CURRENT

"A" – the "device" in which the file is stored (A – RAM, B – Floppy, C and D – ROM disks),

"02" – the type of file (00 and 01 – program names, 02 – DLM capture buffers, 03 – VT100 capture buffers, 04 – DVOM capture buffers, PR – printer files), and

"–CURRENT" – the name of the file.

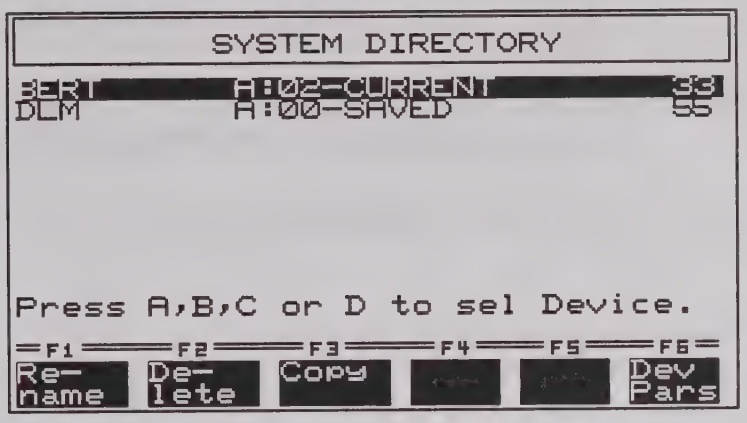


Figure 14-11 INTERVIEW 20 Plus disk System (file) Directory.

(F) INTERVIEW Transfer

To transfer data from a Series 1 INTERVIEW 5/10/15 Plus disk to the INTERVIEW 8000 Series, press the function key labeled INT 10; to transfer data from an INTERVIEW 20 Plus or a Series 2 INTERVIEW 5/10/15/20 Plus disk to the INTERVIEW 8000 Series unit, press the function key labeled INT 20. These keys appear when the cursor is at the top of the Disk Maintenance Screen. The screen then repaints to display the fields shown in Figure 14-12; the **Command** field will display INTERVIEW 10 PLUS TRANSFER or INTERVIEW 20 PLUS TRANSFER for the selection made.

Install the microfloppy from the INTERVIEW 5/10/15 Plus or 20 Plus Series into either of the floppy disk drives. If you are copying the files to another floppy, install a formatted INTERVIEW 8000 Series microfloppy into the other floppy disk drive. (The two types of disks can be distinguished before they are installed as described in Section 14.4(E).)

1. *From.* In this field, select the drive (FLOPPY1 or FLOPPY2) into which you have inserted the disk from the INTERVIEW 5/10/15 Plus or 20 Plus.

2. *Instrument.* This field indicates the type of interface used to record data on the INTERVIEW 5, 10, 15, or 20 Plus. Data from each of the instruments is recorded in a different format, and this must be taken into account in the transfer. There are six possible selections in the **Instrument** field. These are DLM (data line monitor), TERM (asynchronous terminal), X25 MON (X.25 protocol monitor), SNA MON (SNA protocol monitor), BAUDOT (Baudot protocol monitor), and DDCMP (DDCMP protocol monitor).

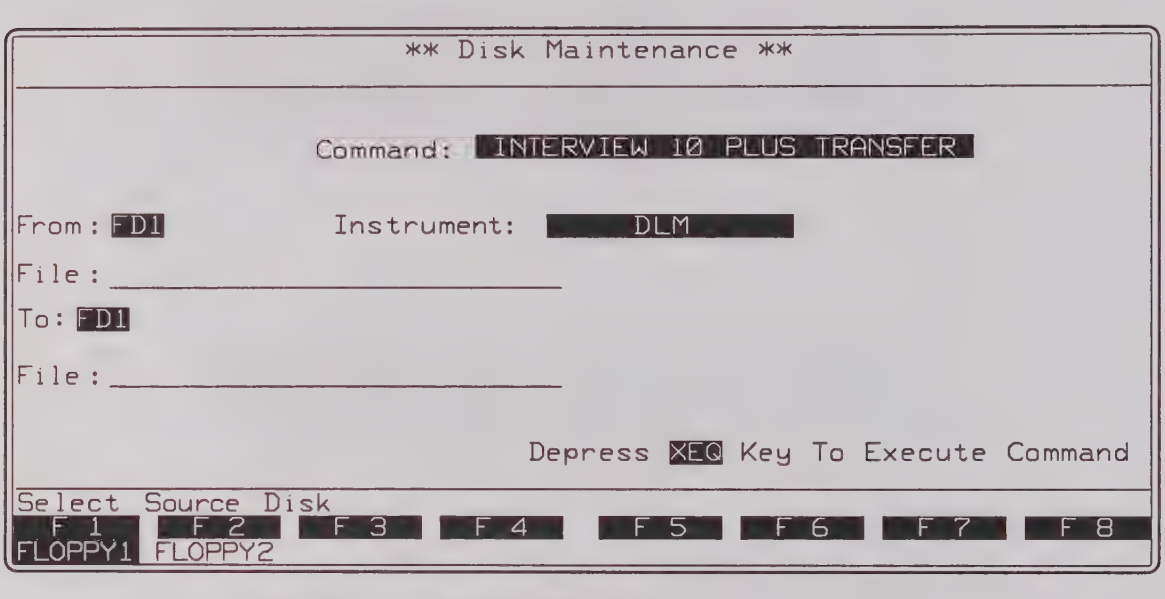


Figure 14-12 Use the Interview Transfer command to transfer data from an INTERVIEW 5/10/15 Plus disk to INTERVIEW 8000 Series disk files.

It is possible to determine from the file listings on the INTERVIEW 5, 10, 15, or 20 Plus what instrument was used to record the data. See Section 14.4(E) for instructions on accessing these file listings.

Once the data is transferred from an INTERVIEW 5/10/15 Plus Series unit, it is stored in the INTERVIEW 8000 Series as ASCII character data, regardless of the instrument on which it was recorded.

However, all INTERVIEW 20 Plus files are transferred as the same character data format in which the data was originally recorded. (For example, data recorded on an INTERVIEW 20 Plus in EBCDIC format would be transferred as EBCDIC data.)

3. *File*. Enter the name of the source file from the INTERVIEW 5, 10, 15, or 20 Plus. The filename for the last data sample collected is –CURRENT (unless the file has intentionally been renamed). Enter this filename exactly as shown. The initial hyphen is part of the name. Other filenames are listed in the disk file directory of the INTERVIEW 5, 10, 15, or 20 Plus. Obtain directory listings as described in Section 14.4(E).

   Only data files can be transferred to the units in the INTERVIEW 8000 Series. For the INTERVIEW 5/10/15 Plus, these files are indicated by the letter D in the second column in the file directory. Do not attempt to transfer program files (marked P).

For the INTERVIEW 20 Plus, these files are indicated by the numbers "02" or "03" after the colon in the second field of the second column on the system (file) directory. Do not attempt to transfer other files. For an explanation on how to distinguish the data files which can be transferred on an INTERVIEW 20 Plus from other types of files, see Section 14.4(E)5.

INTERVIEW 5/10/15 Plus and 20 Plus filenames are always UPPERCASE single words (rather than a full pathname) and must be entered as such in the **File** field in the INTERVIEW 8000 Series.

4. *To*. Indicate in this field the destination drive to which the data is to be transferred. You may transfer data to either of the floppy disk drives (FD1 or FD2) or to the hard disk (HRD).

5. *File*. Enter the destination filename in the second **File** field. You may enter either a full pathname or a pathname relative to the current directory. For a description of how pathnames work in the INTERVIEW 8000 Series refer to Section 15.

   None of the information from the INTERVIEW 5/10/15 Plus file listings is transferred with the file when it is copied into the INTERVIEW 8000 Series. As a result, it is important to use descriptive filenames (or pathnames) to tag your data samples properly.

6. *XEQ*. When all entries on the screen are correct, press [XEQ]. The message "Transfer In Progress" appears on the second line of the screen to indicate that the file is being copied. To abort the transfer while it is in progress, press [F8].

   The message "Transfer Complete" is posted at the top of the screen when the process is complete. If there is a problem with the transfer, an error message appears in the same area. Refer to Appendix A for an explanation of error messages.

   Once the transfer is complete, the name of the file, the file type CHDAT (for character data), the size of the copied file (which is likely to be somewhat larger than the original file), and the date and time of the transfer are recorded in the 8000 Series directory. If the file has been placed in a directory other than the current directory, you must use the Change Directory command to find it in the listings. This command and the File Maintenance screen are described in Section 15.

   Use the regular Data Transfer command to move the data into the data acquisition tracks of the disk for playback and analysis. The Data Transfer command is discussed in Section 14.4(C). Playback is discussed in Section 5.2.

15 File Management

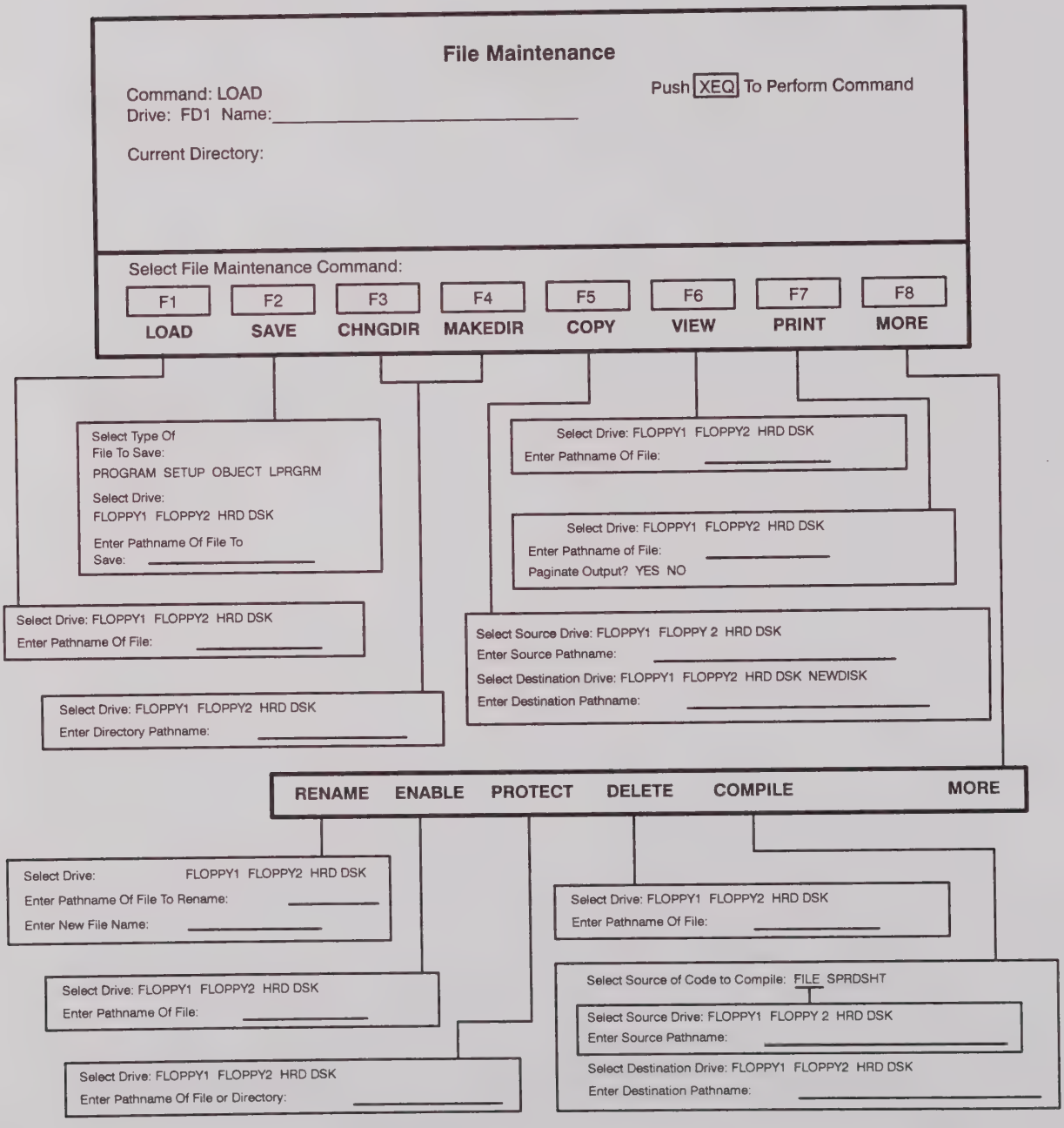


Figure 15-1 File Maintenance menu.

15 File Management

The filing system is a set of files and directories, grouped by disk, created and managed by the user. Files may contain captured line data, setups, protocol package data, entire test programs, or text. Files are identified and accessed by name. The file's location in the system is indicated by its full or relative pathname, as explained later in this section.

All file management functions are performed from the File Maintenance screen (FMAINT; see Figure 15-1 for menu options and Figure 15-2 for default screen). The menu options are described at the end of this section. These functions include copying files, loading and saving files, compiling files or the contents of the Protocol Spreadsheet, permanently deleting files, viewing the contents of certain files, write–protecting or write–enabling files, printing files, and grouping files into directories.

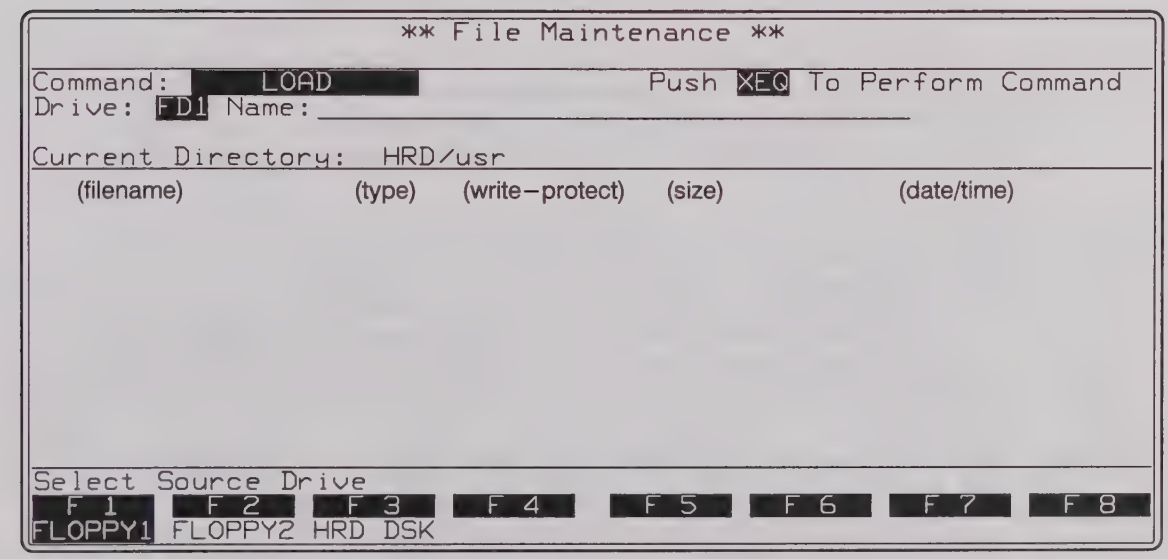


Figure 15-2 Default File Maintenance screen.

15.1 Loading and Saving Files

The File Maintenance screen is an important interface between the filing system and the INTERVIEW's data analysis functions. All INTERVIEW test programs are loaded from the File Maintenance screen. Newly created or modified tests are saved on this menu as well.

NOTES:

1. READ and WRITE commands appear as editing functions on the Protocol Spreadsheet. These commands, described in Section 30, load or save only Protocol Spreadsheet contents and do not interfere with the contents of other menus.

2. Protocol packages, although part of the filing system, are loaded from the Layer Setup screen. For a description of this menu, see Section 8.

15.2 Orienting Yourself in the Filing System

The INTERVIEW's filing system is a simple tree–structured filing system similar to the UNIX™ filing system. (UNIX is a trademark of AT&T Bell Labs.) Each disk has its own hierarchy. At the top of the tree is the "Root" directory, represented by the single slash (/). When new, the System disk contains this directory and two others, */sys* and */usr*. These are subdirectories of the Root directory; that is, they are one step below the top of the hierarchy. (The slash at the beginning of the name locates the directory with reference to the Root directory. Naming conventions for directories will be fully explained later in this section.)

At power–up, the operating system automatically moves you into the */usr* directory of the disk from which the system software was loaded. System software is loaded from Floppy Drive 1 if a system disk is inserted; by next preference, it is loaded from Floppy Drive 2; and, if no other system software is found and a hard disk is present, it is loaded from the hard disk.

(A) Directories

Directories may contain both files and other directories. Once you are in a new directory, the files and subdirectories it contains are listed in the display area of the File Maintenance screen. Directories are indicated by "DIR" in the second column (see Figure 15-3).

1. *How to move through the directories.* You may use the CHNGDIR (Change Directory) command to move into any of these directories and view their contents. As you work within the filing system, you can use this command to change levels and directories. However, you can never be in more than one working directory at a time. This working directory is referred to as the "current directory." The current directory is always posted on the fourth line of the File Maintenance screen.

   To move from the */usr* directory of the boot disk up to the Root directory, use the Change Directory command and enter the pathname /. To move into the */sys* directory, enter the pathname */sys*. There will be no other directories on a new system disk; you must create new directories yourself.

To move from disk to disk, use the Change Directory command and select the correct disk drive in the highlighted window below the command line. In the entry field provided, enter slash (/) to move to the Root directory of the new disk, or enter the absolute pathname of the directory you want. Absolute pathnames are described in the next subsection.

2. *How to create new directories.* New directories are created using the MAKEDIR (Make Directory) command on this menu. A maximum 256 files and/or directories may be created on one microfloppy diskette. The maximum for hard disk for a 16−bit file system is 4,096 with a maximum partition size of 32 MBytes.

   For a 32−bit file system [see Section 14.4(A)5.], the number of files and/or directories proportionally increases/decreases by the multiplier of the partition size. If you double the partition size (32 Mbytes x 2 = 64 Mbytes), the number of files/directories is doubled (4,096 x 2 = 8,192); tripling the partition size (32 Mbytes x 3 = 96 Mbytes) triples the number of files/directories you can have on your hard disk (4,096 x 3 = 12,288); and so on. For the factory standard of 500 Mbytes, the number of files/directories is 64,000.

   Directories follow the same naming conventions as files. These conventions are explained in the following paragraphs. The pathname you enter as you make a directory indicates where you wish to locate the directory in the filing hierarchy. Pathnames are explained later in this section.

```
                    ** File Maintenance **

Command:  LOAD                          Push XEQ To Perform Command
Drive: FD1 Name: _____________________

Current Directory:  HRD/usr
BSC                 DIR          167      12/03/86  12:12
BSC_Test_Data       BITIM     114371      01/30/87  12:30
Emul_3270_Term      PRGM       12620      11/17/86  11:27
Line_Eff_3270       PRGM        4438      11/09/86  17:00
SNA_Bind            PRGM   W    3107      09/06/86  09:11
SNA_Sessions        PRGM   W   10700      09/30/86  10:22
Test_3270_Term      PRGM        9722      11/10/86  07:12
X25_Cert            DIR          213      03/14/87  10:25
X25_Cert_desc       ASCII       1020      03/16/87  19:12
X25_Frm_Lev         SETUP       2504      02/02/87  10:25
X25_Pkt_Size        PRGM        9380      02/22/87  14:32

Select File Maintenance Command
 F 1     F 2     F 3     F 4     F 5     F 6     F 7     F 8
LOAD    SAVE   CHNGDIR MAKEDIR  COPY    VIEW    PRINT   MORE
```

Figure 15-3 Contents of the current directory are listed on the File Maintenance Screen.

(B) Absolute Pathnames

A pathname identifies the path through the directory structure which terminates at a particular file or directory. Each file and directory has an absolute pathname and a relative pathname.

An absolute pathname always begins with a slash. This indicates the path from the "root," or starting point, of the filing system. The directories and/or file it "owns" are listed to the right, each separated from the next higher level by a slash.

The absolute pathname is useful when you aren't sure of your current location in the filing system. Using the absolute pathname always alerts the operating system of the precise location of the directory or file you specify in any command.

(C) Relative Pathnames of Files and Directories

The relative pathname of a file or directory is the route to the file or directory from the current directory. The relative pathname may be as simple as a one–word filename. (It never begins with a slash.) If, for example, your current directory (*dir_c*) contains a file (*your.file*), then the relative pathname of the file is, simply, *your.file*. Now suppose that you move to a different current directory (*dir_b*) which contains *dir_c* as a subdirectory. The same file would have a different relative pathname: *dir_c/your.file*.

Each time you move up a directory level, the relative pathname gains a new subdirectory name as a component. The new component is always separated from its subdirectory (or file) by a slash. The relative pathname of this same file when you are working in the */usr* directory might, for example, be *dir_a/dir_b/dir_c/your.file*. The absolute pathname for the same file would be */usr/dir_a/dir_b/dir_c/your.file*.

(D) Files

Files are the basic elements of the filing system. Files can be identified in the Type field of the Directory Listings. Any type other than "DIR" is a file.

The INTERVIEW files named in the directory listings on the File Maintenance screen are created in a variety of ways. The files in the */sys* directory, for example, are placed on each system disk at the factory. This directory contains data essential to the operating system. It is not advisable to store user files in the */sys* directory.

The */usr* directory, also created on each disk at the factory, is intended for the storage of files (and directories) which you create.

1. *Program files.* Program files, which contain the configuration of all INTERVIEW menus and the Protocol Spreadsheet contents, are created when you use the SAVE command on this menu.

2. *Setup files.* Use the SAVE command on the File Maintenance screen to create Setup files. Setup files contain a partial set of configured menus.

3. *Object files.* Object files, which contain the configured Setup menus, the Layer packages, and the object–code compilation of the Trigger menus and the Protocol Spreadsheet, are created when you use the SAVE command on the File Maintenance menu.

4. *Linkable–object files.* Use the Compile command on the File Maintenance menu to create linkable–object (LOBJ) files. Linkable–object files may contain the object–code compilation of standard C code. The compiled C code in linkable–object files usually contains the definitions of user–created routines.

   Linkable–object (LOBJ) files may also contain the compiled contents of the Protocol Spreadsheet, different from object (OBJ) files which additionally contain all menu configurations.

5. *Linkable–program files.* Use the Save command on the File Maintenance menu to create linkable–program (LPGM) files. Linkable–program files contain all of the setup menus of program (PRGM) files except for Triggers and source–code Spreadsheet. In addition to the menus, LPGM files contain the linkable–object code compilation of the Protocol Spreadsheet. When you load an LPGM file and go to the Protocol Spreadsheet, all you see is an OBJECT block identifier that references the pathname of the LPGM file.

   Once a linkable–program file is loaded (via the File Maintenance Load command), you may modify the setup menus and rerun or save the program. Since the spreadsheet program has already been compiled, changes to the setup menus do not cause a lengthy recompile time.

6. *Protocol Spreadsheet files.* Use the WRITE command provided by the spreadsheet editor (described in Section 30), specify a filename, and press [XEQ] or [RETURN] to save only the contents of the spreadsheet. The file which results is listed in the File Maintenance directory as type ASCII; however, when you want to use the file again, it should be read in by the spreadsheet editor (rather than loaded from the File Maintenance screen.)

7. *Data files.* Use the Data Transfer command on the Disk Maintenance screen to store data from disk into a file. See Section 14.4(C). The files which result appear in the directory listings on the File Maintenance screen. Before you replay data files, you must transfer them back onto the data acquisition tracks of a disk; a file is not a valid **Source** of data on the Line Setup menu. Do not attempt to load data files from the File Maintenance screen.

(E) Naming Conventions for Files and Directories

There are a few simple rules to keep in mind when naming files. The maximum length of any component is 12 characters. Legal characters for filenames are letters; numbers (0–9); and the symbols dash (—), underscore (_), dollar sign ($), and period (.). Filenames must start with a letter or a period. Upper and lower case letters may

be used. Filenames are case–sensitive; that is, the pattern of upper and lower case letters in an existing name must be repeated exactly any time that file is referenced. Wildcards may be used in some operations, as described in Subsection (F), below.

Filenames may include extensions for convenience, but the system attaches no special meaning to the extension. When you refer to a file with an extension, you must include the extension in the filename.

Filenames cannot be duplicated within the same directory. If, for example, there is already a file in your current directory called *unifile* and you save a newly created file to the name *unifile*, then the old file is destroyed.

Single periods (.) and double periods (..) have special uses when referring to files or directories. Their uses are described in the paragraphs which follow.

(F) Use of Wildcards in Filenames

There are several ways in which you can designate files using wildcards to be acted upon by a chosen command. Table 15-1 lists the legal wildcard definitions in the INTERVIEW file system.

Table 15-1
Wildcards With File Designations

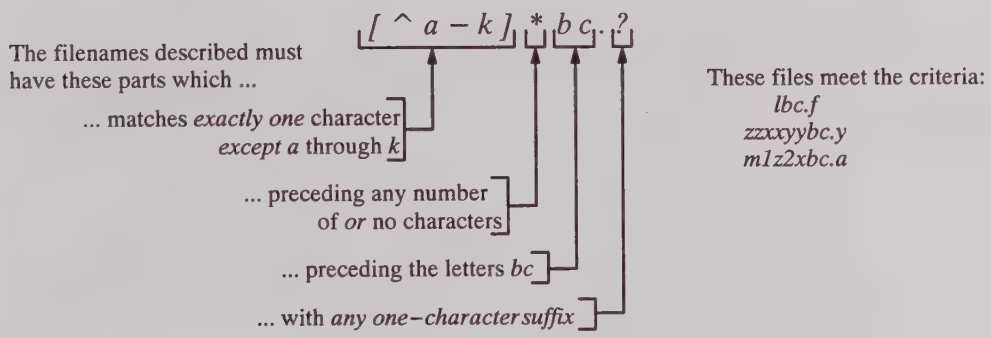
Wildcard	Meaning
*	matches **0 or more** of any character
?	matches any **single** character
[ ]	matches any **one** of the **listed** characters, i.e., **[–abcdK23457]**; ranges are allowed: **[–a–dK2–57]** [1]
[^]	matches any **one** character **except** those listed, i.e., **[^cdef045678–]**; ranges are allowed: **[^c–f04–8–]** [1]

[1] The hyphen (–) is used as a wildcard match or exclusion in two ways:
(1) To designate an inclusive range of characters, it is used between characters.
(2) To designate *itself*, it must be listed either at the beginning or ending of the sequence.

A space ( ) will be read as a space. Spaces are not used in filenames, so neither should they be used within wildcard bracket designations.

As illustrated in Table 15-1, the hyphen (–) is used two ways within brackets: as the hyphen character, it is placed at the beginning or end of the sequence; or, used between two characters, it designates inclusive ranges.

For example, using the wildcard designations found in Table 15-1, the file designation *[^a–k]*bc.?* describes one or more files which have the following attributes:



1. *Limitations.*

   Wildcards are special characters that do pattern matching and can only be used when the cursor is in the menu area.

   Wildcards *cannot* be used for the MAKEDIR operation.

   A destination file with wildcard designations must *only* match a single file.

   A destination directory with wildcard designations must *only* match a single directory.

   Wildcard symbols may be used to type in "shorthand" filenames with only one match; these are operated upon as if the real pathname had been entered. If problems occurred during the operation, errors messages are those displayed as if a single pathname had been entered for the filename.

2. *Marked files.*

   Generally speaking, marked files and wildcard–designated files and directories are mutually exclusive, with *one* exception: you may copy a marked file into a *single destination* file or directory whose "shorthand" name includes wildcard symbols, as long as the name identifies *one and only one* file or directory. Multiple files may also be marked to be copied to a single destination directory named in the same manner. Like other operations with marked files, the cursor must remain in the file list area.

   If errors occur, the operation may have been successfully completed upon *some or none* of the marked files; those which encountered errors will remain marked. We recommend you try the operation again with each of the remaining marked files, one at a time; this permits the system to report a more specific message for each file.

(G) Use of Periods in Pathnames

Single periods and double periods, when used alone as file components, have special meanings. A single period (.) always represents the current directory. A double period (..) always represents the parent directory; that is, the directory immediately above.

The double period (..) is a useful abbreviation for defining pathnames. A common use for the component would be to change the current directory to the parent directory. In this case, you don't need to use the name of the parent directory. Just select the Change Directory command and enter .. as the pathname to the new current directory. String a series of these together, separated by slashes (for example, Change Directory ../../..) to move up the file tree.

Study the file system in Figure 15-4 to see how the following points apply.

1. The relative pathname of *your.file* is *dir_c/your.file*. Its absolute pathname is */usr/dir_a/dir_b/dir_c/your.file*.

2. The relative pathname of *other.file* is *dir_d/other.file*. Its absolute pathname is */usr/dir_a/dir_b/dir_d/other.file*.

3. To move up to the */usr* directory from the current directory, enter the pathname ../.. when you use the Change Directory command.

4. To move from the current directory (*dir_b*) to *dir_e*, use the Change Directory command and enter this pathname to the new directory: *../dir_e*. (This is simpler than using the absolute pathname, */usr/dir_a/dir_e*.)

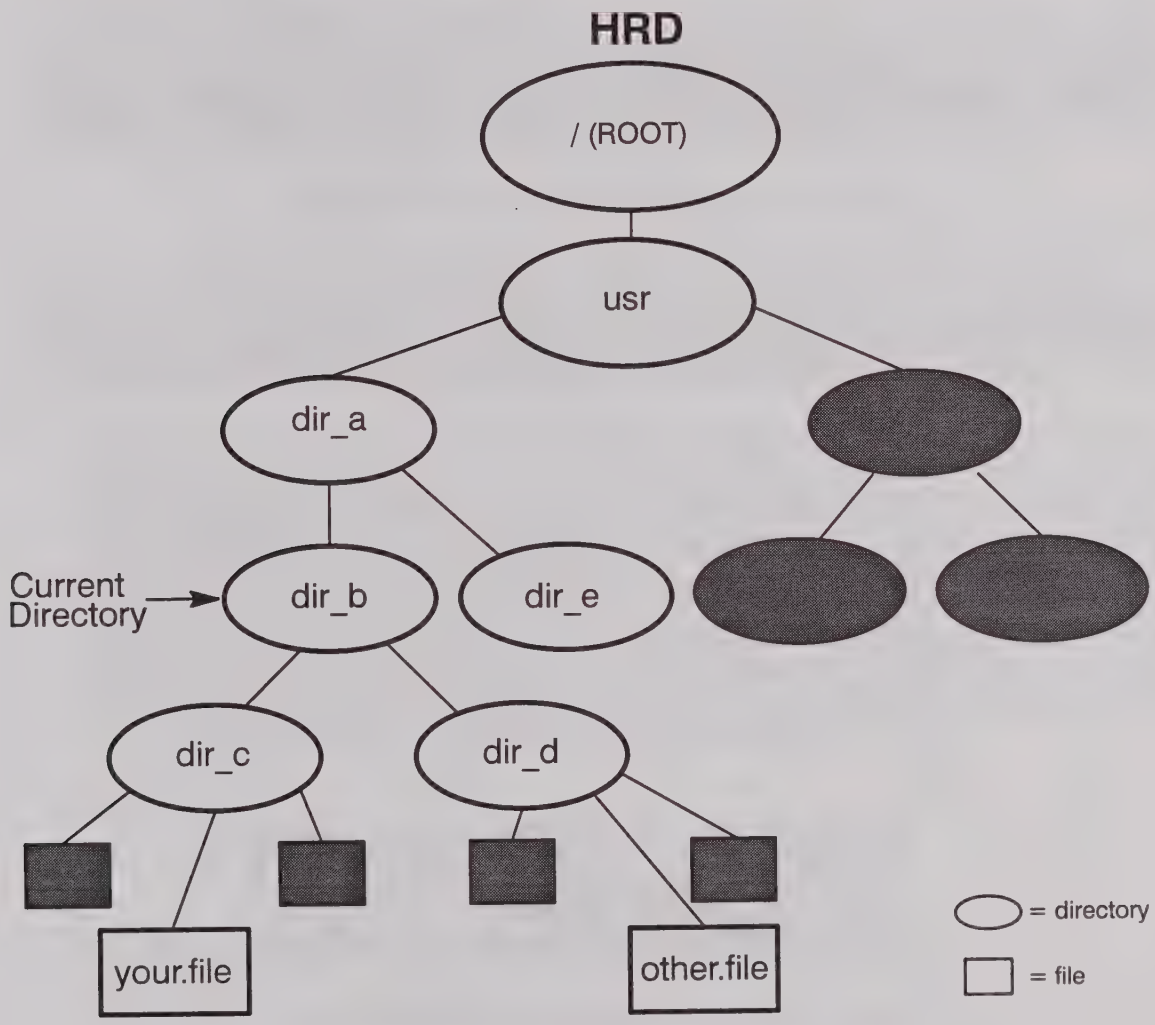


Figure 15-4 The INTERVIEW has a hierarchical filing system.

15.3 The File Maintenance Screen

(A) The Top of the Screen

Twelve commands can be executed from the File Maintenance screen (see Figure 15-1 for a full set of menu fields). There are two alternate banks of commands. Use the MORE (F8) function key to switch from bank to bank (see Figure 15-5 and Figure 15-6). When you press the function key to select a particular command, the screen repaints, and a set of unique menu fields appears. The selected command always appears on the top line of the menu.

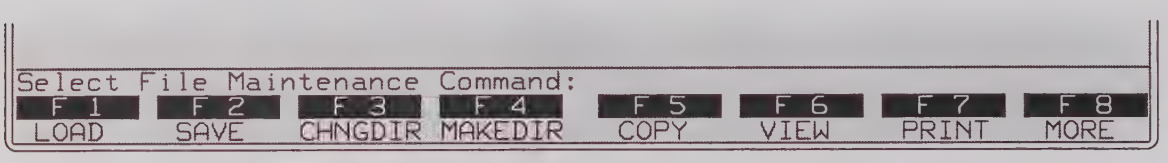


Figure 15-5 Press *MORE* to display the alternate bank of function keys.

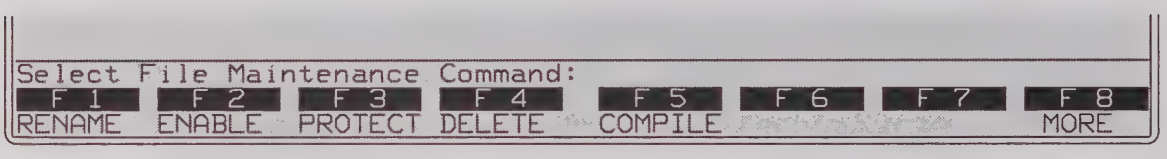


Figure 15-6 Alternate bank of function keys.

1. *Current drive*. At the top of the screen, just below the **Command** field, the name of the current disk is given (see Figure 15-7). The files displayed in the directory listings on the screen reside on the disk named in this area.

   NOTE: The current disk at power–up is the disk from which the INTERVIEW is initialized. If all three drives are present, the INTERVIEW checks FD1, then FD2, and finally HRD for system initialization software and boots from the first system disk it finds.

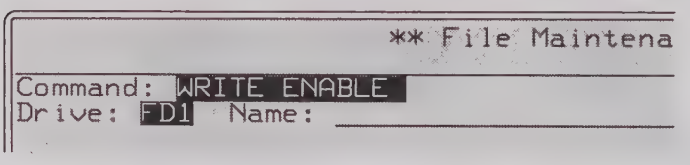


Figure 15-7 The name of the current disk is selected in the *Drive* field.

2. *Name field*. Every command requires at least one filename and provides a field in the menu area for filename entry (see Figure 15-7) where you MAY enter the name of the file to be acted upon. The file's absolute or relative pathname can be entered. If you enter the filename, you must type it exactly as it appears in the directory listings.

   NOTE: There are several ways to specify the file or files which are to be acted upon. Read Section 15.3(C) before executing any command.

   Study the directory listings if you are uncertain of the filename. If you require more information on a file, use the View command to see the contents of the file.

(B) Directory Listings

The name of the current directory is listed in a field at the top of the screen. The contents of the current directory are listed in a 12–line field in the center of the menu. Six columns of information appear for each file (see Figure 15-8). The (relative) filename is given in the first column.

The Type field to the right defines the contents of the file. All types that may appear in this column are defined in Table 15-2. When a file is write–protected, the letter "W" appears immediately to the right of the Type column.

The size of the file (in bytes) is given in the next column. The date and time that the file was last modified appear in the two far–right columns.

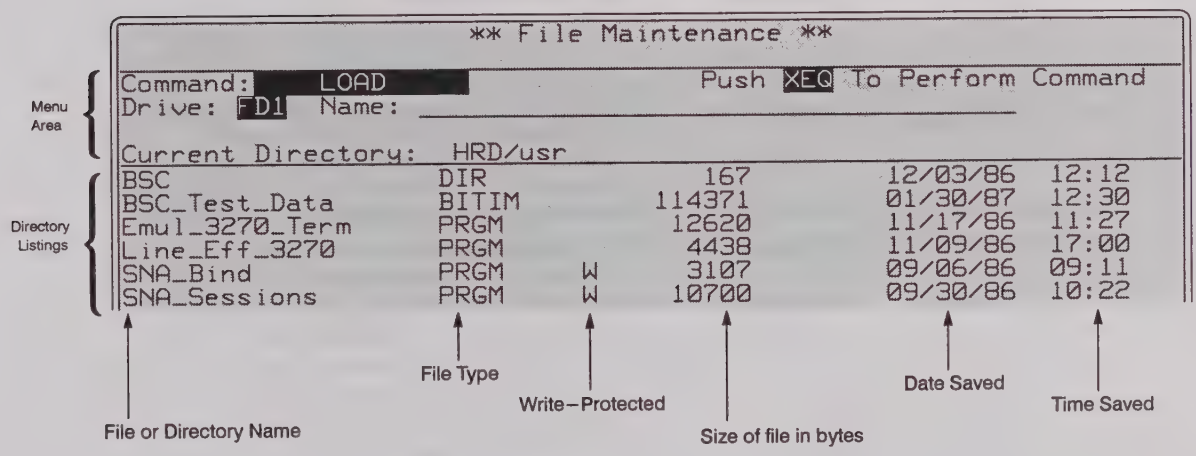


Figure 15-8 Current directory listings. Data types are listed in the column to the right of the file or directory name.

1. *Moving through directory listings.* The contents of the current directory are displayed on the File Maintenance screen.

   More than 12 lines of information may be available for the directory. To display additional lines, position the cursor on the last line of the listings and press ↓. (This will display new lines one at a time.)

   Press SHIFT and ↓ at the same time to move to the end of the listings. Press NEXT PAGE to display the next 12 lines of information. Position the cursor on the first line and press ↑ to expose previous lines one at a time. Press PREV PAGE to display the previous 12 lines.

   Press SHIFT and ↑ to move to the top of the listings.

   To see the listings for another directory, you must use the Change Directory command.

Table 15-2
File Types

Type	Meaning	Comments
PRGM	Program	Full set of configured screens.
SETUP	Setup	Subset of five configured Setup screens, in particular, excludes Trigger Setup screens, Layer Setup screen, and Protocol Spreadsheet.
OBJ	Object	Five configured Setup screens, the Layer packages, and the object–code compilation of the Trigger menus and the Protocol Spreadsheet
LOBJ	Linkable Object	Object–code compilation of a standard C file or of the contents of the Protocol Spreadsheet. C files may be generated via the WRITE/U (editor)command on the Protocol Spreadsheet.
LPGM	Linkable Program	Setup menus of a PRGM file (except for Triggers and source code of spreadsheet) in addition to the linkable–object code compilation of the Protocol Spreadsheet. LPGM files are compatible between units with different hardware and/or software configurations.
BITIM	Recorded data	Bit–image data saved for reanalysis. May be reformatted and rechecked for BCC error.
CHDAT	Recorded data	Character data saved for reanalysis. This data has already been formatted and checked for BCC error (if received block–checking enabled) and may not be reformatted or rechecked.
ASCII	Text	Text file generated by user on Protocol Spreadsheet screen.
DIR	Directory	A directory, potentially containing files and subdirectories, which is part of the structured file system.
SYS	System files and internal INTERVIEW files	Files generated by General Signal Networks – Mount Laurel, including Personality packages.

(C) Selecting Files

There are several ways in which you can designate files to be acted upon by a chosen command.

1. *Name field.* The first means of selecting a file is to enter its name in the **Name** field at the top of the menu.

   NOTE: The cursor must remain in the menu area (see Figure 15-8) in order for the command to operate on the named file.

2. *Cursor location*. You may designate a single file to be acted on by moving the cursor into the directory listings and positioning it over the desired file. The file name will be highlighted.

   NOTE: When the cursor is located in the Directory Listings, the highlighted file will be acted upon by any command executed. Any file named in the **Name** field will NOT be acted upon.

   If you have selected a file with the cursor, but the file–maintenance command shown at the top of the screen is not the one you want, press [DONE]. The cursor will return automatically to the **Command** field. Choose the appropriate command and press [DONE] again. The cursor will return to its previous location, highlighting the selected file.

3. *Marking files*. For certain commands, the [MARK] key can be used to select files on which you want the command to operate. To mark a file, locate the cursor over it and press [MARK]. Once a file is marked, you need not fill in the **Name** field for the command.

   The next command executed will operate on the marked file or files. Even if there is a filename in the **Name** field, marked files (and NOT the entered file) will be acted upon.

   Marked files also take precedence over the file at the cursor location. That is, if files are marked, they will be acted upon by the command, but the file at the cursor location will not be acted upon.

   With one exception, [MARK] works only with commands which are used to operate on multiple files simultaneously. These commands are Copy, Delete, Write–Protect, Write–Enable, and Print. The exception is the Compile command. In this case, only a single file should be marked. If you mark more than one file for compiling, the following error message will be displayed: *"Too many source files selected."*

   Wildcard symbols may be used during an operation with one or more marked files. If errors occur, the operation may have been successfully completed upon *some or none* of the marked files; those which had errors will remain marked. We recommend you try the operation again with each of the remaining marked files, one at a time; this permits the system to report a more specific message for each file.

4. *Unmarking a file*. [MARK] is an alternate action key. To deselect a marked file, locate the cursor next to it and press [MARK] again.

   Once the command is executed, the file is no longer marked. Marked files are no longer marked if you leave the File Maintenance screen.

(D) Executing Commands

To execute any of the 12 commands, make sure that all entries are correct. Then press [XEQ]. Status messages will inform you of any errors. Refer to Appendix A for an explanation of error messages.

(E) Load

For fast access to this command, press [LOAD] as an alternative to accessing the File Maintenance screen with function keys. Use this command to load a working copy of a file from disk into the internal memory of the INTERVIEW. The Load command is frequently used to load complete programs into the INTERVIEW in preparation for running a test. Any program or setup file you wish to modify must first be loaded.

Only program, setup, object (OBJ), or linkable–program (LPGM) files can be loaded with this command.

- Program files are a full set of configured menus, including the Layer Setup screen, Trigger Setup screens, and the Protocol Spreadsheet.
- Setup files are a smaller set of configured menus which includes only the five Setup screens: Line Setup, Interface Control, BCC Control, Front–End Buffer Setup, and Bit Error Rate Test Setup.
- Object files are the configured Setup menus, the Layer packages, and the object–code compilation of the Trigger menus and the Protocol Spreadsheet.
- Linkable–program (LPGM) files contain all of the setup menus of program (PRGM) files except for Triggers and source–code Spreadsheet. In addition to the menus, LPGM files contain the linkable–object code compilation of the Protocol Spreadsheet. When you load an LPGM file and go to the Protocol Spreadsheet, all you see is an OBJECT block identifier that references the pathname of the LPGM file. Although it makes no difference from which drive you load a linkable–program file, in order for the program to run, the file must remain accessible on that same disk drive.
- Once a linkable–program file is loaded, you may modify the setup menus and rerun or save the program. Since the spreadsheet program has already been compiled, changes to the setup menus do not cause a lengthy recompile time.

> **NOTE:** To successfully execute the File Maintenance Copy command on linkable–program files, you must have software revision 8.00 (or later). LPGM is an unknown file type to earlier software revisions.

You must specify the disk on which the file you are loading resides and the (relative or absolute) pathname of the file before executing the load command. The simplest means of loading a file from the current directory is to place the cursor over the filename in the directory listings and press [XEQ]. Cursor selection overrides any entries that may appear at the top of the menu.

NOTES:

1. Remember that loading a program or setup overwrites the program or setup already in the INTERVIEW. Save the resident program or setup if you wish to use it later, and then load another file.

2. Loading an object file allows you to enter Run mode without recompiling, unless you make substantive changes (i.e., changes to menus and fields not listed in Table 2-2).

3. The spreadsheet portion of a program (when saved via the Protocol Spreadsheet WRITE editor command) can be loaded via the READ command without overwriting the contents of other menus. See Section 30 for an explanation of the READ and WRITE commands.

4. Protocol layer setups are loaded from the Layer Setup screen (see Section 8). They may also be loaded along with a program, linkable–program, or object file.

5. Data files are not "loaded" from the File Maintenance screen. Instead, see Section 14.4(C), Data Transfer.

6. Linkable–object (LOBJ) files are not loaded from the File Maintenance screen. Depending on the file's contents, use either the OBJECT block–identifier or the *#pragma object* preprocessor directive to access the compiled C code in an LOBJ file. Refer to Sections 28.4 and 55.4.

7. You cannot load the Printer Setup file (*/sys/print_setup*) from the File Maintenance screen. It is loaded automatically during boot–up. See Section 2.1(B).

(F) Save

For fast access of this command, press SAVE as an alternative to accessing the File Maintenance screen with function keys. Use this command to preserve a newly created program, setup, or object file or to retain any changes you have just made to an existing file. If saving to an existing filename, the existing file *must* be the same file *type* as the file you are saving to overwrite it.

Before executing this command, you must select the type of file to be saved. Program, setup, object (OBJ), and linkable–program (LPRGRM) are the only options on this screen.

- A setup is a set of five configured menus: Line Setup, Interface Control, BCC Control, FEB Setup, and BERT Setup.

- A program contains all menus, including the Layer Setup screens, Trigger Setup screens, and the contents of the Protocol Spreadsheet.

- An object contains the configured Setup menus, the Layer packages, and the object–code compilation of the Trigger menus and the Protocol Spreadsheet. If you have not compiled the program prior to saving the object, the compilation occurs as part of the SAVE process.

- Linkable–program (LPGM) files contain all of the menus of program (PRGM) files except for Triggers and source–code spreadsheet. In addition to the menus, LPGM files contain the object–code compilation of the Protocol Spreadsheet. The compilation occurs as part of the SAVE process.

  When you Save a program as an LPGM file, your spreadsheet program *excluding* C *#pragma*s is saved as compiled linkable–object code. To retain *#pragma*s in your program, reference them as the hook text in a *#pragma hook 0* before you save the program. Although the *#pragma hook 0* is not saved, the hook text is.

  There are two exceptions to this rule: *#pragma object* and *#pragma hook 0.* Enter these directives on the spreadsheet following the formats explained in Section 55, C Basics. They will be saved intact in LPGM files.

  To include other *#pragma*s, *#pragma il_buffers 128* for example, in an LPGM file, enter them in a *#pragma hook 0* as follows:

  #pragma hook 0 "#pragma il_buffers 128"

  NOTE: To successfully execute the File Maintenance Copy command on linkable–program files, you must have software revision 8.00 (or later). LPGM is an unknown file type to earlier software revisions.

Before executing the command, you must also select the drive to which you wish to save the file, and you must enter the pathname to the file you are saving. If you are saving to a filename which already exists in the current directory, you may indicate the file by placing the cursor over its name in the directory listings before pressing [XEQ]. Cursor selection of a file overrides any entries which may appear at the top of the menu. The existing file *must* be the same file *type* as the file you are saving to overwrite it.

You cannot save to a directory which does not already exist. First, create the directory with the Make Directory command.

Press [XEQ]. During the Save operation, status messages are posted at the top of the File Maintenance screen. For a diagnostic message about errors that prevent compilation of OBJ or LPGM files, press [PROGRAM], SPDSHT, [EDIT], GO-ERR. Press GO-ERR again for a message about the next error. Continue until no more errors are detected.

You may abort the Save procedure by pressing the ABORT softkey or [PROGRAM]. Note, however, that if the destination file is an existing file, it may have been partially overwritten.

NOTES:

1. You may use the Save command to create a new set of default values for all menus. If you create a file called */usr/default* on the initialization disk, all menus will be set to your saved selections when the INTERVIEW boots from that disk. See Section 2.2(B) for a description of this procedure. See Section 14.1 for disk selection at boot-up.

2. If you only wish to save the contents of the spreadsheet, use the Protocol Spreadsheet WRITE command (see Section 30). Or use the File Maintenance Compile command to save the compiled object-code version of the contents of the spreadsheet. See Section 15.3(P).

3. Saving an *object* or *linkable-program* file requires a considerable amount of disk space and should be reserved for frequently used tests with long compilation times. (The program file for a test may occupy only 5320 bytes, while the object file for the same test occupies 109894 bytes.)

4. Preserve a copy of the source code (the program version) of a test as well as the OBJ or LPGM version. The program code is more *versatile* than the OBJ file code: in subsequent software revisions, the program version may still compile even if the object-code version no longer does. You may then generate a new object file from the source-code version.

   Although linkable-program files are as compatible as program files, editing on the Protocol Spreadsheet must take place in the source program file.

5. Data can also be saved to a file using the Data Transfer command on the Disk Maintenance screen; see Section 14.4(C). Once data is saved in a file, the file appears in the directory listings of the File Maintenance screen.

6. Linkable-object (LOBJ) files are created via the Compile command, not the Save command. See Section 15.3(P).

7. Save the Printer Setup configuration file (*/sys/print_setup*) from the Printer Setup screen, not the File Maintenance screen. See Section 16.3.

(G) Change Directory

This command, labeled CHNGDIR on the function key, is the only method of moving from one current directory to another. Indicate the drive on which the new directory resides and give its absolute or relative pathname before executing the command. The name of the new current directory replaces the old one on the fourth line at the top of the File Maintenance screen, once the command is executed.

NOTES:

1. By definition, all relative pathnames of files and directories change once you change to a new current directory.

2. To move up to a higher directory, you must name the directory in the **Name** field.

3. If the name of the parent (the next higher) directory is not known, enter two periods (..) in the **Name** field. This moves you up one directory.

4. To change to a directory on another disk, select the drive in the rotating window and enter the absolute pathname of the directory in the name field.

(H) Make Directory

You must use the MAKEDIR (Make Directory) command to create a new directory. Use this command to create your own file hierarchy. You may create any number of subdirectories to the root (/) or */usr* directory on any disk, and you may subordinate directories to another directory, thereby building as many levels of subdirectories as you wish.

The relationship of one directory to another is identified by its name, as explained earlier in this section.

> **NOTE:** When you create a directory, you may use the absolute pathname for the new directory to locate it anywhere you specify on any of the disks. If you use a relative pathname, the new directory will be created as a subdirectory of the current directory.

(I) Copy

Use this command to copy a file or group of files to a new location. (The original copy or copies will remain, unless you choose to delete them using the Delete command.) This command is useful, for example, when you wish to copy certain files from one disk onto another disk. If you are using only one floppy disk drive, the system will prompt you to insert the "source" disk or the "destination" disk at the proper times and you should select NEW as your destination disk on the File Maintenance screen.

NOTE: To successfully execute the File Maintenance COPY command on linkable–program files, you must have software revision 8.00 (or later). LPGM is an unknown file type to earlier software revisions.

When you select the Copy command, you will be asked to select the source drive (where the file or files to be copied now reside), enter the source pathname of the file to be copied, select the destination drive, and enter the destination pathname (the intended directory location and filename for the new copy). The pathname you enter may be relative or absolute.

Wildcards can be used to name source files in a COPY operation; wildcard symbols are discussed in Section 15.2(F). Multiple source files can also be specified by marking several files in the file list area.

Table 15-3 illustrates the twelve possibilities for copying both single and multiple files and directories. The table assumes the directories and files are write–enabled.

Table 15-3
Source and Destination Copy Guidelines

	Source \ Destination	File	Directory	Non–existent
Single:	File	Overwrite existing file	Create file into directory; overwrite existing file [1]	Create destination file [2]
	Directory	***ERROR***	Existing files of same name overwritten	Create destination directory [2]
Multiple:	Files	***ERROR***	Existing files of same name overwritten	***ERROR***
	Directories	***ERROR***	Create destination directories [3]	***ERROR***

[1] Example: Copying the file */usr/programs/x25* to the writeable directory */usr/save* actually copies the file into */usr/save/x25*. If */usr/save/x25* already exists on the specified drive, it is overwritten; if it does not exist, it is created.

[2] Target will be created, but only if all preceeding parent directories exist on the destination drive and the last *one* parent directory is writeable. Example: If destination file is to be */usr/programs/x25*, then the writeable directory */usr/programs* must exist on the drive where the destination file is to be created.

[3] Source directories are copied into subdirectories of the destination directory on the specified drive; these destination subdirectories must have the same name as the corresponding source directory.

1. *Copying a group of files.* Multiple filenames may be "marked" in the directory for copying. See Marking Files, Section 15.3(C)3.

   If you designate a directory as the source you want to copy, all files and subdirectories contained in the directory are also copied.

   Wildcard symbols may be used to copy a group of files; wildcards are discussed in Section 15.2(F). If wildcard symbols are used to copy a group of files, the destination filenames begin with the first wildcard, appended to the destination parent name.

   For example, copy multiple files from */HRD/x25/*/x/*k** to the directory */FD1/bkup*. If the actual source files copied are
   /HRD/x25/A/x/aka and */HRD/x25/B/x/back*
   then the new destination filenames would be
   /FD1/bkup/A/x/aka and */FD1/bkup/B/x/back*

   NOTE: Copying a group of files to a floppy disk drive may require more space than is available on one disk. When the first destination floppy disk is full, you will be prompted to insert the next disk.

   If you are copying a group of files from one place to another on the *same* floppy disk, you must make certain you have enough space left on the disk to hold all of the files you wish to copy. This same advice holds for copying a group of files from any disk drive to the hard disk drive: verify you have enough space on the destination disk to hold the copied files.

2. *Copying an entire directory.* You may indicate an entire directory to be copied. The directory and all its subdirectories and files will be copied. If the name of an existing directory is given, all files and directories are placed inside that directory. But the original directory is not retained. This means that any files or directories in the destination directory are, unless write–protected, overwritten by files or directories of the same name in the copied directory. You may also copy files to a fictitious subdirectory. As long as the subdirectory has a real parent directory, such a destination pathname is valid. (For example, if you have a directory named */usr/programs* on FD1, you may copy a directory from the hard disk to */usr/programs/x25* on FD1, even though the directory */x25* did not previously exist.) The subdirectory is created as part of the Copy command, and all files and directories from the source are copied into it.

3. *Copying an entire disk.* If both floppy disk drives are operable, you would use the Duplicate Disk command on the Disk Maintenance screen. See Section 14.4(D).

   If you have only one floppy disk drive available, the Duplicate Disk command is not applicable to your hardware. However, it is possible to copy an entire disk to another disk in three steps: format the destination disk, copy the root directory to the disk, and transfer the data acquisition tracks to the disk.

After a disk is formatted, it may be copied to via the Copy command. Copy the root directory from one disk to another; the file name for both the source and destination disks will simply be "/" and the softkey selection for the destination disk is NEWDISK. The system will prompt you to insert the "source" disk or the "destination" disk at the proper times. You must then use the Data Transfer command on the Disk Maintenance screen—see Section 14.4(C)—to copy to data acquisition tracks to your destination disk. The disk is now duplicated.

> **NOTE:** Copying to a new disk in the same drive will use the same memory as the current program and the Layer Personality packages. The procedure may need fewer repetitions of disk–insertion/removal if it is done at a time when a large program and multiple layer packages are not loaded in.

(J) View

The purpose of the View command is to allow you to look at the contents of a file without actually loading the file. The program or data currently in the internal memory of the INTERVIEW is not overwritten by the View command.

Prior to executing the View command, you must select the drive on which the file resides and provide the (relative or absolute) pathname of the file.

When you press [XEQ], the screen repaints, displaying the first 12 lines of the file in the Directory Listing area. New function keys are available as labeled.

Descriptive text is presented to the user when a program or setup file is viewed. For an ASCII file, the file is presented "as is." Bit–image and character–data files cannot be viewed.

1. *Moving through a view file*. The beginning of a file appears on the screen when a file is viewed. The cursor location is highlighted. Use [↓] to move the cursor to the bottom of the screen and continue pressing [↓] to scroll down through the file a line at a time. Move the cursor to the top of the screen by repeatedly pressing [↑] and continue pressing it to scroll back up through the file.

   Press [SHIFT]–[↓] to move down to the end of the file. Use [SHIFT]–[↑] to move to the top of the file.

   Use [NEXT PAGE] to move one page (12 lines) forward in the file. Press [PREV PAGE] to move backward one page in the file.

   To leave the file being viewed and return to the File Maintenance screen, press [DONE] or the function key labeled ABORT.

(K) Print

This command prints a hard copy of the file when a printer is properly connected to the INTERVIEW's Printer Port. Indicate the disk on which the file resides, as well as the file's relative or absolute pathname. Then make a selection in the **Paginate** field.

If you select **Paginate:** YES , each printed page will contain a header. The header includes the name of the file (not its pathname) and the page number:

File : filename Page : 1

Section 16 provides instructions on configuring a printer for use with the INTERVIEW and explains printed format.

(L) Rename

Use the Rename command to assign a new name to a file. A file can only be renamed if it resides on the active disk (named on the current directory line). In order to rename a file, you must indicate the disk on which the file resides, provide the pathname of the file (that is, give the old filename), and enter the new name you wish to give the file.

> **NOTE:** When you rename a file, the original file disappears.

The Rename command can also be used to move a file to a new location. Both new and old locations must be on the active disk (named on the current directory line.) To relocate a file to a different disk, use the Copy command, and then delete the original file.

> **NOTE:** Do not use the abbreviations (. and ..) for naming the new location for a renamed file since they are valid characters for a file name. They will be interpreted as part of the new name.

(M) Enable

This command is used to remove the write protection from a file or directory. Once a file or directory has been write–enabled, the save, delete, and rename commands can be executed on the file or directory.

When a directory is write–enabled, the files within the directory are also enabled, unless they have been individually write–protected. In this case, you must also enable each file before it can be modified or deleted.

Prior to executing the command, you must select the drive on which the file or directory resides and provide the (relative or absolute) pathname of the file or directory. Or you may simply cursor–select a filename for write–enabling and press XEQ. Multiple filenames may be "marked" in the directory for write–enabling. See Marking Files, Section 15.3(C)3.

(N) Protect

This is the Write Protect command. It is used as a security measure. When a file is write–protected, any attempt to save, delete, or rename the file will be rejected.

Prior to executing the Write Protect command, you must indicate the drive on which the file (or files) reside and you must provide the name of the file (or files) you wish to protect.

A directory may be named, cursor–selected, or marked for write protection. This write–protects the individual files within the directory. It is possible to write–protect the individual files as well; in which case, both the directory and the file must be enabled before the file can be modified or deleted.

> **NOTE:** Reformatting a floppy disk destroys all files and data, whether they are protected or not. Floppy disks can be write–protected manually against formatting, as explained in Section 1. The hard disk cannot be write–protected against a Format command entered from the Disk Maintenance menu.
>
> It is strongly recommended that you regularly back up the hard disk as well as any floppies you wish to preserve.

(O) Delete

Use this command to remove files permanently from a disk. Prior to executing the Delete command, you must select the disk drive from which you are deleting the file and give the (relative or absolute) pathname of the file you wish to delete. In order to delete an entire directory, you must first delete all files and subdirectories it contains. To delete a subdirectory you must be located in the directory above it (the parent directory).

> **NOTE:** Be aware that you can delete a file by placing the cursor over a filename in the directory listings and then executing the Delete command. Cursor selection takes precedence over any filename in the **Name** field.
>
> You may also mark one or more files for deletion with MARK. Any marked files take precedence and will be deleted instead of the file at the cursor location or a file named in the field at the top of the menu.

(P) Compile

Use this command to compile and save as object code the contents of the Protocol Spreadsheet or a file.

> **NOTE:** If you want to save setup menus in addition to the compiled spreadsheet program, use the Save command and create a linkable–program (LPGM) file. See Section (F) above.

One of the advantages of using linkable–object (LOBJ) files instead of object (OBJ) files is that OBJ files created on one unit will not run on another unit that is configured differently. In contrast, LOBJ files (as well as LPGM files) are

transparent to the configuration of the unit—i.e., they are just as transparent as the code would be, were it actually present in the spreadsheet buffer. The code, however, must still be compatible with the various menu parameters. For example, a call to *ll_il_transmit* or a SEND action, whether contained in an LOBJ file or written to the Protocol Spreadsheet, would not be compatible with a Line Setup selection of **Mode:** MONITOR .

Linkable–object files also assist the programmer in efficiently using the INTERVIEW's memory and spreadsheet buffer. Refer to Section 28.4(D).

1. *File.* FILE is the default selection. A valid source file for compiling contains only standard C code, typically definitions of user–created routines. It may also contain *#pragma hook 0* preprocessor directives. (Refer to Section 55.4.) The code does not have to be a complete program. Program or setup files are not valid source files.

   NOTE: Write C code from the Protocol Spreadsheet to a file by using the WRITE/U spreadsheet–editor command. See Section 30. The file will be type ASCII in the directory listing on the File Maintenance screen.

   If you choose FILE , also select the source drive (where the file resides) and enter the source pathname of the file to be compiled. To select a source file, you may enter the relative or absolute pathname of the file, position the cursor over the filename in the current directory, or mark the file in the current directory. See Selecting Files, Section 15.3(C). Note that the new file will only overwrite a file of the *same file type*.

   Depending on whether or not the file contains "hook", use either the OBJECT block–identifier (Section 28.4) or the *#pragma object* preprocessor directive (Section 55.4) on the Protocol Spreadsheet to access the C code in the file.

2. *Spreadsheet.* If you choose SPREADSHEET , the contents of the Protocol Spreadsheet will be compiled. The spreadsheet program may include C code, softkey–generated entries, or a combination of the two, but it must be a valid program in order to compile.

   The linkable–object file which results will always contain system–generated *#pragma hook* directives, at least one of which will be a type–zero hook. (See Section 55.4.) To access this spreadsheet file, therefore, you must reference it on the Protocol Spreadsheet with the OBJECT block–identifier. See Section 28.4.

Whether you are compiling a file or the spreadsheet, specify a destination drive and enter the destination pathname (the intended directory location and filename for the LOBJ file). The pathname you enter may be relative or absolute.

Linkable−object files follow the standard naming conventions discussed in Section 15.2(E). As an added convention, you may want to append the suffix *.o* to the end of the name.

> **NOTE:** A */lib* subdirectory (if present in the */sys* or */usr* directory) is included in the search routine for linkable−object files. See Section 28.4. We recommend, therefore, that you make a */usr/lib* directory for storing the LOBJ files you create.

Press [XEQ]. During the Compile operation, status messages are posted at the top of the File Maintenance screen. If there are errors in the source file that prevent compilation, the following message will be displayed: *"Compilation failed — Errors detected."* For a diagnostic message about the first error, press [PROGRAM], SPDSHT, [EDIT], GO−ERR. Press GO−ERR again for a message about the next error. Continue until no more errors are detected.

You may abort the Compile procedure by pressing the ABORT softkey or [PROGRAM]. Note, however, that the destination file may have been partially overwritten if compile was to an existing file.

16 Printer Control

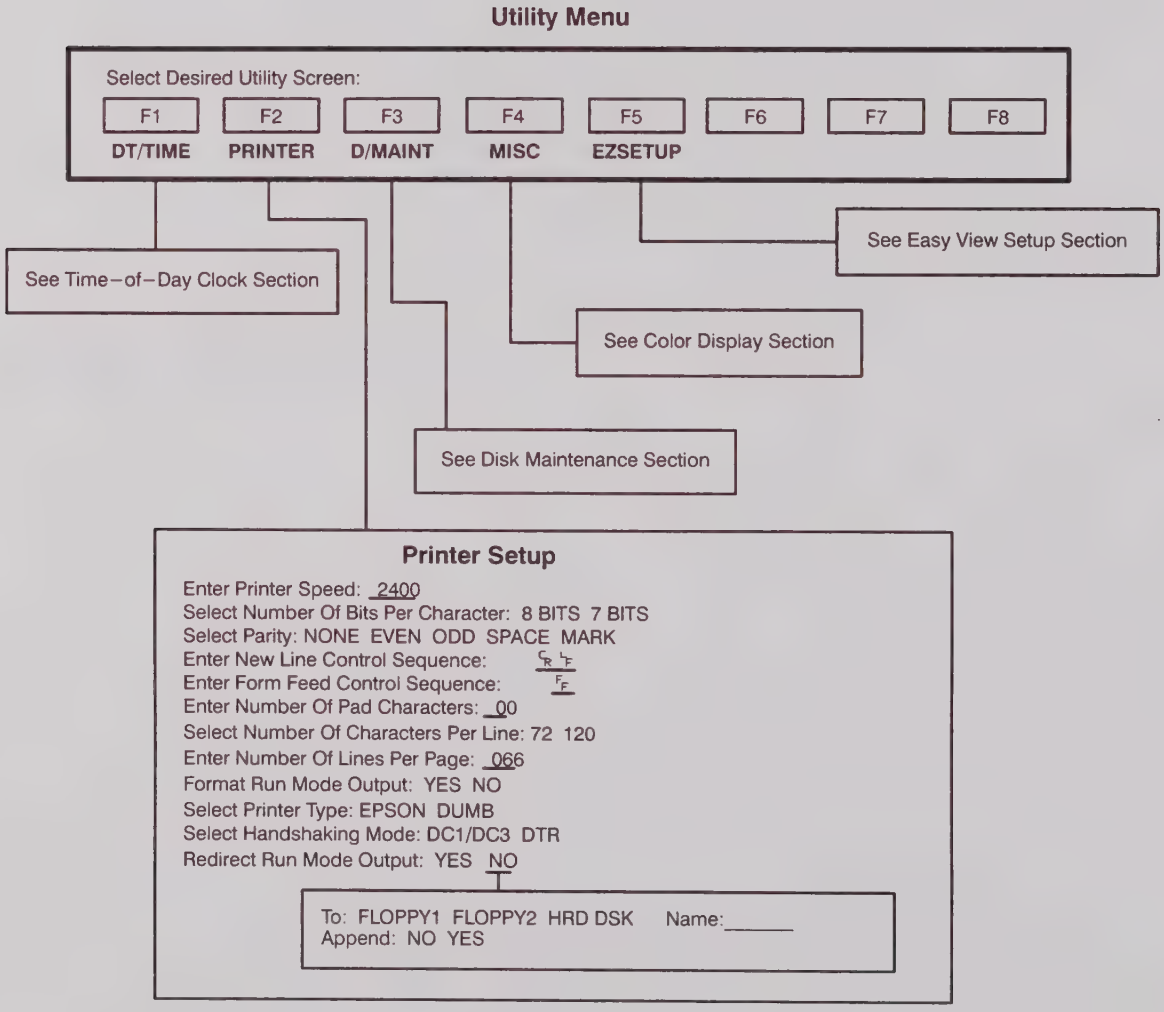


Figure 16-1 The Printer Setup menu.

16 Printer Control

The INTERVIEW will control most serial ASCII printers. In Program mode you may print program menus, triggers, the spreadsheet, and the contents of most files. In Freeze mode, you may print character data, statistics, and protocol and program traces. You may print prompts and statistics in Run mode (real−time.) Printing is controlled from the keyboard, from the Protocol Spreadsheet, and from the Printer Setup screen (see Figure 16-3), described later in this section.

16.1 The Connector

A male RS−232/V.24 Printer connector is located on the rear of the INTERVIEW (see Figure 16-2). The Printer connector is a DCE interface: it transmits to the printer on Pin 3 (RD) and applies ON voltage (+12 V) on Pins 5 (CTS), 6 (DSR), and 8 (CD). Information can be transmitted asynchronously in 7− or 8−bit ASCII code, with selectable parity.

For flow−control information, see Section 16.2(K).

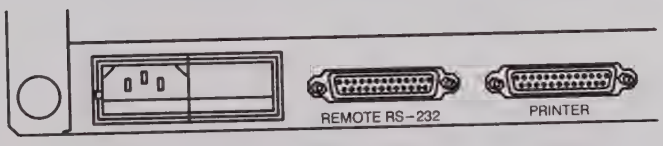


Figure 16-2 The printer connector is a 25−pin (DB−25) male connector.

16.2 Configuring the Printer Setup Screen

The Printer Setup screen is shown in Figure 16-3. The various menu fields are described in this section.

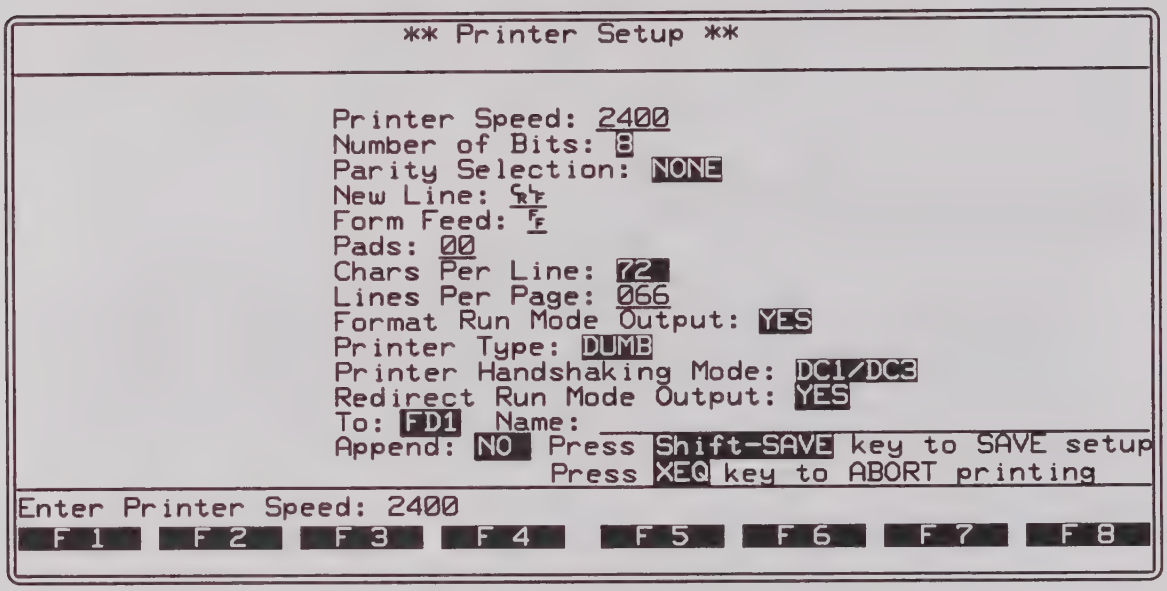


Figure 16-3 Printer Setup screen.

(A) Speed

Enter the speed of the printer in this field. Default entry is 2400 bps. Maximum print speed is 19200 bps.

(B) Number of bits

Select the number of data bits (i.e., total bits less parity) which the printer expects in each byte. Possible selections are `8` (the default) and `7`.

(C) Parity

Select the parity of the printer. Available options are **Parity:** `NONE`, `EVEN`, `ODD`, `SPACE`, and `MARK`. The default selection is `NONE`.

(D) New Line

In this field, enter the control character or characters which the printer interprets as "Start a new line." Alphanumeric, control, and hexadecimal characters are legal entries. Default new-line characters are ␍␊ (Carriage Return/Line Feed). If the field is cleared and left blank, no new-line characters will be sent to the printer.

(E) Form Feed

Enter the printer's Form Feed control sequence (the control characters which cause the printer to advance to the top of the next page). Alphanumeric, control, and hexadecimal characters are legal in this field. You may enter up to four characters. The default entry is ^F (press CTRL−L or HEX, 0, C; do not use HEX, F, F). If the field is cleared and left blank, no form−feed characters will be sent to the printer.

(F) Number of Pads

Some printers require pad characters following a new line command so that they have time to return to the start of the line and advance the paper without losing data. Following each new line command, the INTERVIEW adds the number of nonprinting pad characters specified in this field. Only numeric entries from 1 to 99 are legal. The default entry is 0.

(G) Characters Per Line

Select the number of characters to be printed per line. Available options are 72 (default) and 120. If your printer's line length is 119 characters or less, choose 72. Lines longer than 72 characters will wrap to the next line.

If your printer's line length is 120 or greater, choose 120. Lines longer than 120 characters will wrap to the next line.

(H) Lines Per Page

Enter the length of your printer's page (in lines). The default is 66 lines. Numeric values from 1 to 999 are legal in this field.

(I) Format Run Mode Output

When YES is selected (this is the default), the Character or Screen Buffer is formatted before it is printed. (The data in the buffer is not affected.) The formatted buffer allows three positions per character and assigns symbols to special characters. If NO is selected, the character buffer is transmitted to the printer "as is" in single−line display with ASCII coding. **Format Run Mode Output:** NO might be appropriate if you were monitoring data that had already been formatted for a printer.

(J) Printer Type

Select the type of printer attached to the INTERVIEW. The options are DUMB and EPSON.

EPSON formats the menu screens so that the printouts approximate the INTERVIEW screen. The default selection, DUMB, places a carriage return after each field on the menu.

(K) Printer Handshaking Mode

The INTERVIEW responds to XON/XOFF control characters sent by the printer. The control character DC3, interpreted by the INTERVIEW as XOFF, causes transmission of data to the printer to be suspended. When the INTERVIEW receives the control character DC1 (interpreted as XON), it begins transmitting data to the printer again.

Select DTR as the handshaking mode, and the INTERVIEW also responds to DTR status sent by the printer. When DTR is dropped, transmission to the printer is suspended.

(L) Redirect Run Mode Output

The default selection is NO. Select YES to write real–time prompts, counters, and timers to a disk file. To redirect Run–mode output, the **Source** field on the Line Setup menu must be LINE and the **Initial Condition** field on the Record Setup menu must show NOT RECORD. Otherwise, the Line and Record Setup menus will override a **Redirect Run Mode Output:** YES selection.

When a spreadsheet PRINT action is executed, redirected output is captured on disk together with a time stamp indicating date, hour, and minute. (You cannot use this selection to redirect buffer output generated in Freeze mode.) If you use the C print routines discussed in Section 63, the output will not be time–stamped.

> **NOTE:** Output can be redirected to a file while data is being recorded in bit–image RAM. Recording to disk, however, overrides the **Redirect Run Mode Output:** YES selection. Once you activate recording (to disk) via the RECORD key, the spreadsheet RECORD ON action, or the *start_rcrd_play* C routine, output will be sent to the printer port. Even if you suspend recording, output will continue to be directed to the printer port.

When YES is selected, additional entry fields will be displayed on the Printer Setup Screen

To: Enter the destination drive. Selections are FLOPPY1, FLOPPY2, and HRD DSK. The default is FLOPPY1.

Name: Enter the relative or absolute pathname of the ASCII file to which the output will be written. This can be an existing file or a new file name. If the designated file doesn't exist on the destination disk, an ASCII file by that name will be created.

Append: Default selection is NO. Use this selection when you wish to overwrite the data on an existing file. Select YES to append the data to the end of an existing file.

NOTE: The **Append:** NO selection will cause the permanent loss of data in the existing file. If you wish to save the data already in a file, use **Append:** YES or create a new file name.

Table 16-1
Character Representations on Menu Screen Printouts

Plasma Display	Printout
Special characters: F ($^{7}_{E}$ flag), M (bit mask), X (don't care) (( )) S (sync)	Character representation: \7E \BM \DC \(\) \:S
\ (backslash)	\\
Not Equal (bar through character). Example: ≠	\NE character. Example: \NE\7E
Control characters. Examples: $^{S}_{Y}$, $^{E}_{X}$, $^{S}_{X}$	Upper–case mnemonic (displayed as a backslash followed by two upper–case characters). Examples: \SY, \EX, \SX
Hexadecimal characters. Examples: $^{5}_{4}$, $^{B}_{2}$, $^{F}_{F}$	Displayed as a # followed by two printed characters. Examples: #54, #B2, #FF

16.3 Saving the Printer Setup

If there is a Printer Setup configuration that you frequently or always use, the INTERVIEW can boot–up with that menu configuration.

Press SHIFT–SAVE to save the configured Printer Setup menu to a file. The resulting ASCII file is called *print_setup* and is located in the */sys* directory of the boot drive. During boot–up, only the */sys* directory on the boot drive is searched for this file. If it exists, the configured Printer Setup is automatically loaded. Otherwise, the INTERVIEW loads the default Printer Setup.

NOTE: Each time you save the Printer Setup, the contents of *print_setup* are overwritten. Rename or write–protect this file if you want to save its contents.

If you subsequently load a program (PRGM), object (OBJ), or linkable–program (LPGM) file, the Printer Setup is not affected since these file types do not include the Printer Setup menu.

You can also load *print_setup* from the Printer Setup menu. Press [SHIFT]–[LOAD]. Loading this file does not affect any other INTERVIEW menu screen.

16.4 How to Print Static Displays

While displayed, most menus can be printed individually via the [PRINT] key. (See Figure 16-4.) Some menus, however, are summary screens for a group of submenus. Pressing [CTRL]–[PRINT] when these menus are displayed produces a printout of the group of submenus. Symbols for menu screens and the special characters they represent are listed in Table 16-1.

(A) Printing The Set of Program Menus

Display the Program Menu screen and press [CTRL]–[PRINT] to print all the Program menus listed, including the Protocol Spreadsheet. (Trigger menus will not print if they have not received programming entries.)

(B) Printing the Setup Menus

Press [CTRL]–[PRINT] with the initial Setup screen displayed to obtain a set of print–outs for all Setup screens (Line, Display, BCC, Front End Buffer, and Interface Control).

(C) Printing Triggers

Display the Trigger Summary screen. Press [PRINT] to print the summary and [CTRL]–[PRINT] for as many of the trigger menus as have received programming entries. To print an individual trigger, display it and press [PRINT].

(D) Printing The Protocol Spreadsheet

Display the Protocol Spreadsheet and press [CTRL]–[PRINT] to print the entire contents of the Spreadsheet. Press [PRINT] alone to print only that portion of the Spreadsheet which is visible on the screen. The header of every printed page will include software and firmware revision levels as well as page number as follows:

```
S/W v10.00 ROM v7.00   INTERVIEW 8800   2/29/92   15:02   Page: 1
```

You may insert a form–feed command to the printer anywhere on the Protocol Spreadsheet by employing the following string:

```
/*F_F */
```

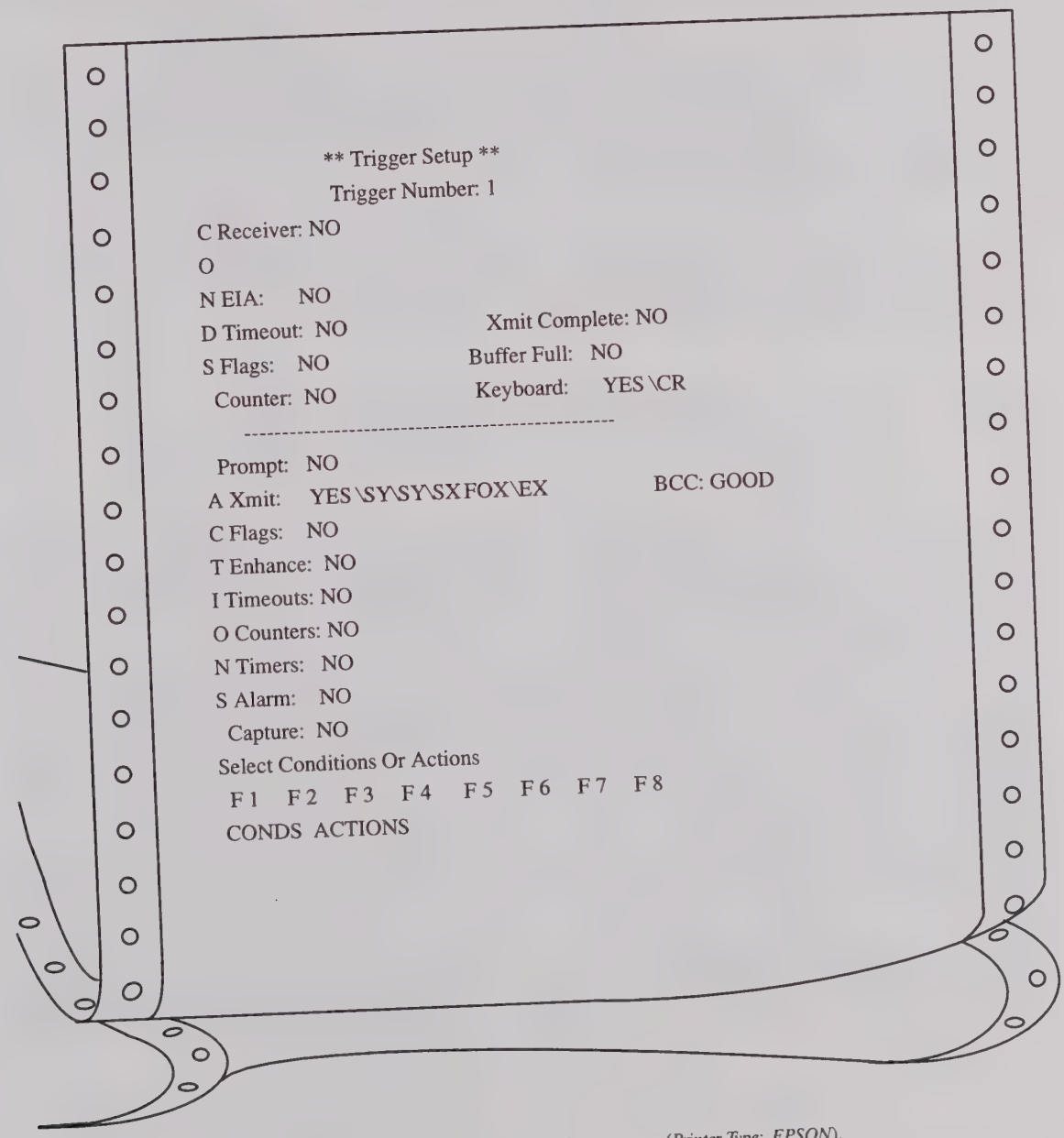


Figure 16-4 Printout of the Trigger Setup screen (*Printer Type: EPSON*).

If you have a reverse–video block on the spreadsheet (editor mode), press PRINT to print only that block, no matter how long or short.

This string uses the $^{F}_{F}$ character formed by the keys CTRL–L, not HEX, F, F. The string will send a form–feed instruction to the printer.

(E) Printing the Layer Setup Screen

Display the main Layer Setup screen and press CTRL – PRINT to obtain printouts of the main screen and all layer–specific setup screens for layer packages that are loaded in.

16.5 How to Print Data

Line data, protocol traces, user traces, program traces, and the contents of the Display Window can be printed in Freeze mode. Both graphical and tabular statistics can be printed in Freeze mode or in Run mode (real–time.)

Table 16-2
Character Representations on Data Printouts

Plasma Display	Printout
Alphanumeric characters (all ASCII noncontrol except space)	Character is preceded and followed by a space
Fill symbol (▒)	Displayed as three dots (...)
Space	Displayed as a blank space followed by two letters **sp**
Special characters F (7E flag), G (good BCC), B (bad BCC) A (abort) S (sync)	Character representation: [F] [G] [B] [A] [S]
Control characters Examples: 5Y, 5X, 5X	Lower–case mnemonic (displayed as a blank space followed by two lower–case characters). Examples: sy, ex, sx
Hexadecimal characters Examples: 54, B2, FF	Displayed as a blank space followed by two printed characters. Examples: 54, b2, ff
Lead–state symbols ‾ (lead state high) _ (lead state low) ⌝ (lead transition high to low) ⌜ (lead transition low to high)	Character representation: + – \– /+

(A) Printing Line Data

Line data can be set up to print in single−line, dual−line, or data−plus−leads formats. Setup information, date, time, offset, percentage, and page number are printed at the top of each page. The date and time are the current system date/time when printing data from an actual line; for recorded data, the date/time of recording is displayed. The symbols for data printout and the characters they represent are listed in Table 16-2.

Press [PRINT] to print data which is frozen on the screen. Data is formatted according to the selection made on the Printer Setup screen—see Section 16.2(I). Figure 16-5 shows the frozen dual−line data screen; Figure 16-6 shows the data printouts at 72 and 120 characters per line.

Press [CTRL]−[PRINT] to print the entire data buffer from the frozen screen. The data buffer will continue to print until the end of the data or until you exit and reenter Run mode (press [PROGRAM], [RUN].)

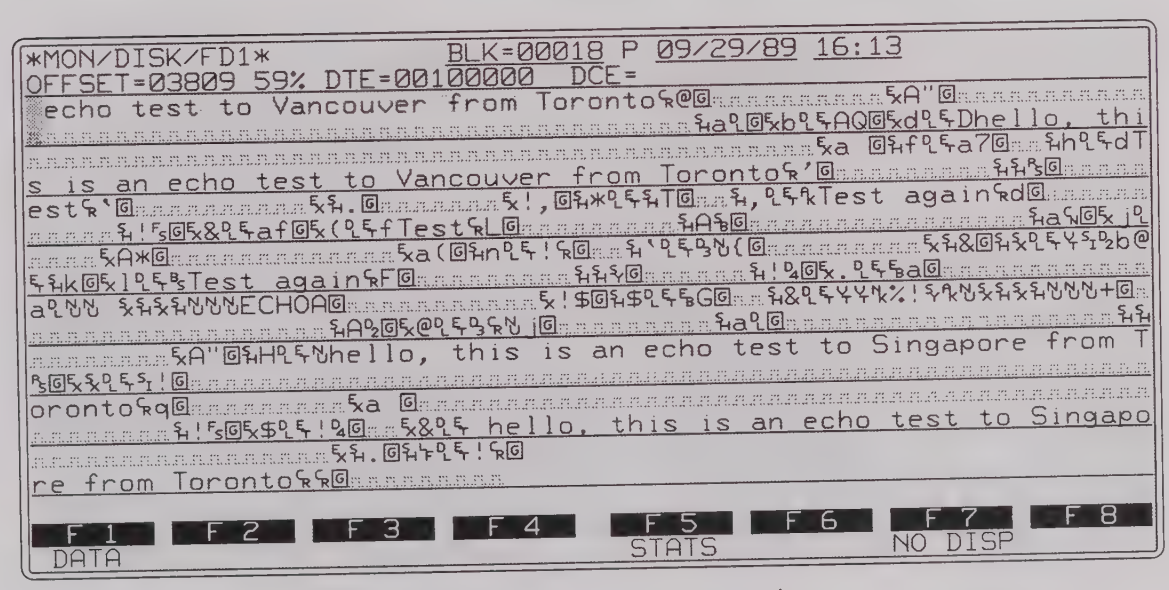


Figure 16-5 Data screen in Freeze mode.

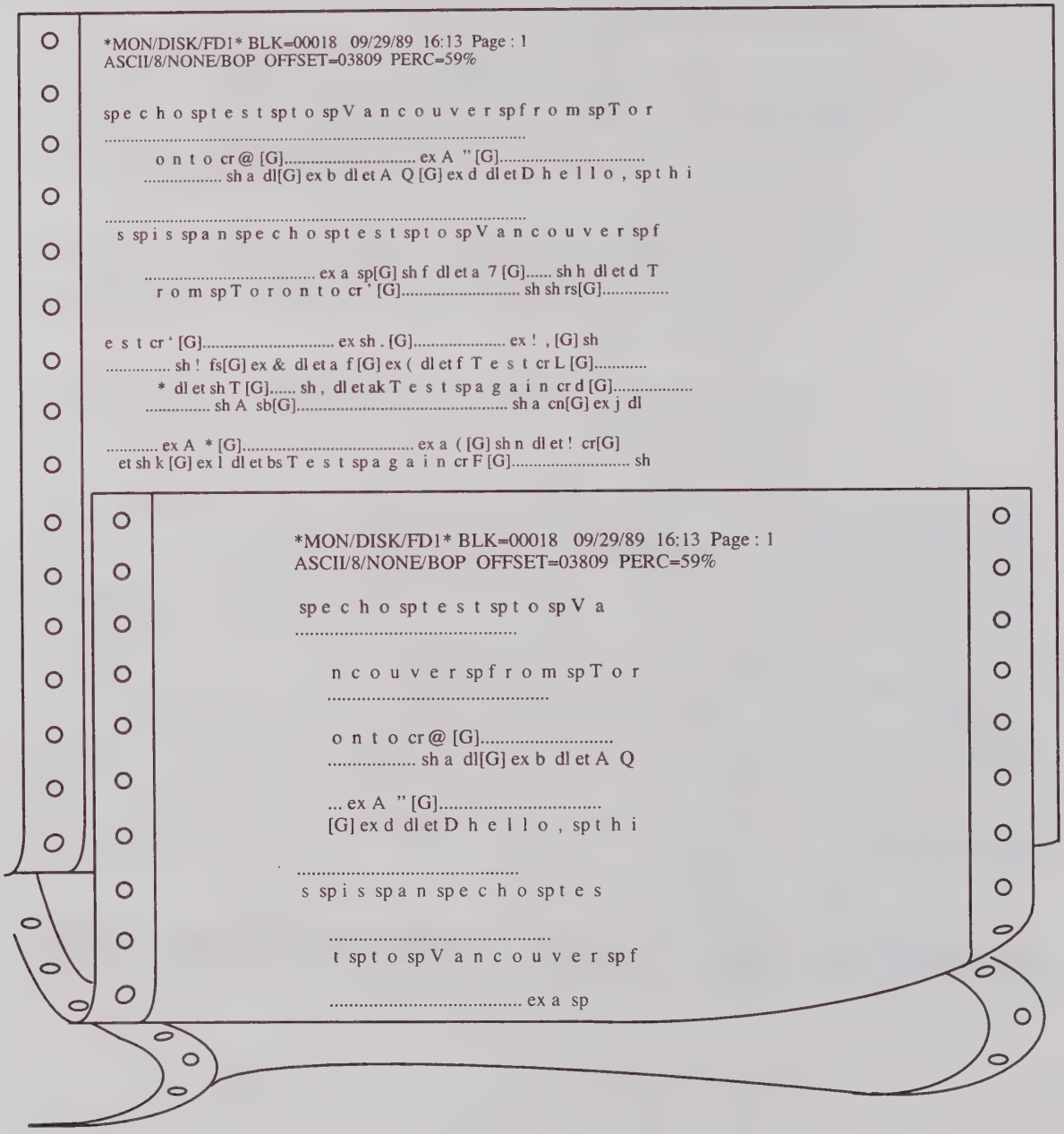
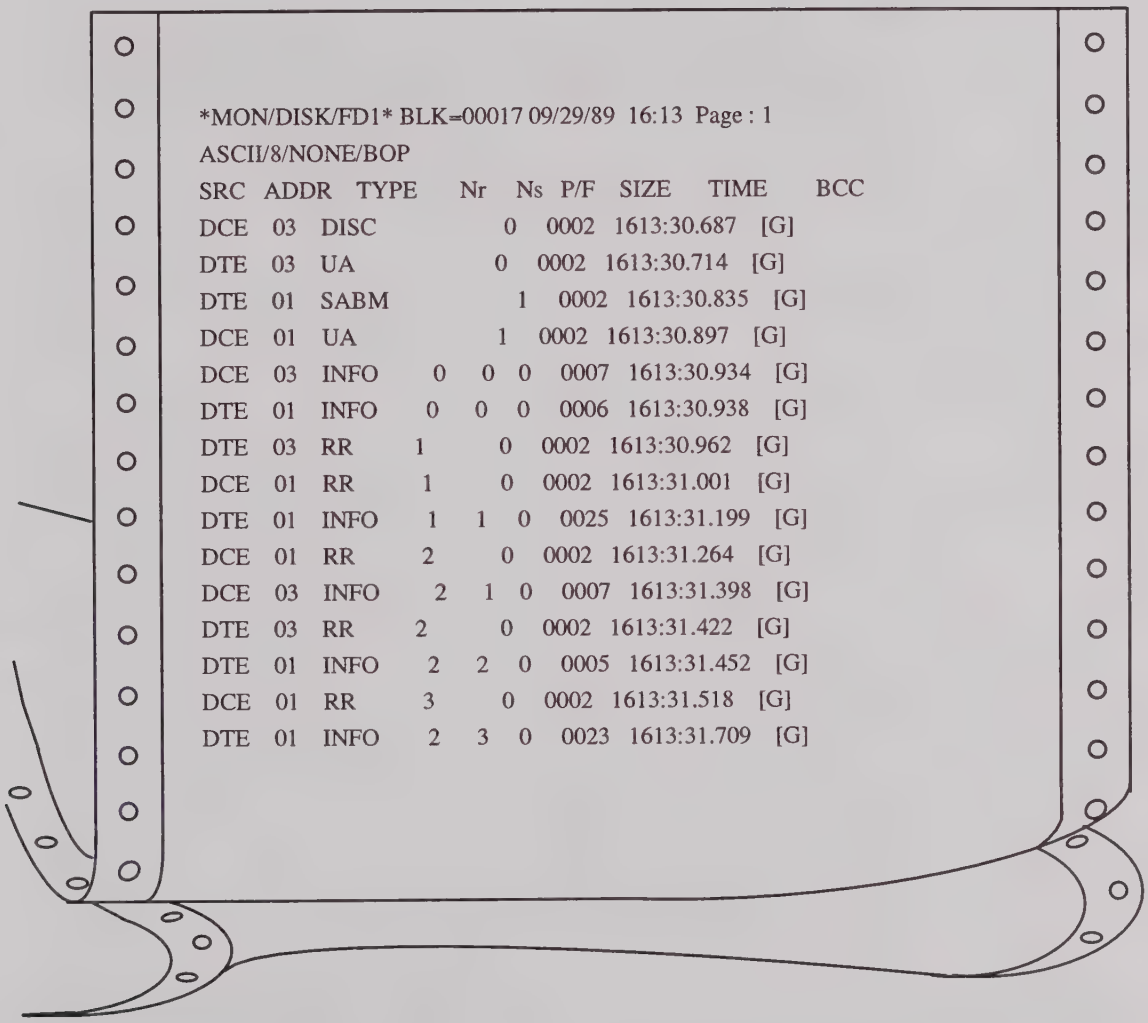


Figure 16-6 Printout of the data display in Figure 16-5 at 120 (top) and 72 (bottom) characters per line.



```
*MON/DISK/FD1* BLK=00017 09/29/89  16:13  Page : 1
ASCII/8/NONE/BOP
SRC  ADDR  TYPE     Nr   Ns  P/F   SIZE    TIME       BCC
DCE   03   DISC              0    0002  1613:30.687   [G]
DTE   03   UA                0    0002  1613:30.714   [G]
DTE   01   SABM              1    0002  1613:30.835   [G]
DCE   01   UA                1    0002  1613:30.897   [G]
DCE   03   INFO      0       0    0     0007  1613:30.934   [G]
DTE   01   INFO      0       0    0     0006  1613:30.938   [G]
DTE   03   RR        1       0         0002  1613:30.962   [G]
DCE   01   RR        1       0         0002  1613:31.001   [G]
DTE   01   INFO      1       1    0     0025  1613:31.199   [G]
DCE   01   RR        2       0         0002  1613:31.264   [G]
DCE   03   INFO      2       1    0     0007  1613:31.398   [G]
DTE   03   RR        2       0         0002  1613:31.422   [G]
DTE   01   INFO      2       2    0     0005  1613:31.452   [G]
DCE   01   RR        3       0         0002  1613:31.518   [G]
DTE   01   INFO      2       3    0     0023  1613:31.709   [G]
```

Figure 16-7 Layer 2 Protocol Trace printout in Freeze mode.

(B) Printing Protocol Traces

Protocol traces can be printed in Freeze mode via the PRINT key. Press PRINT to generate a printout of the current screen (see Figure 16-7). Press CTRL – PRINT to print the entire buffer from the frozen screen. Printing will be aborted if you leave Freeze mode.

(C) Printing Program Trace

The Program Trace can be printed in Freeze mode via the PRINT key. Press PRINT to generate a printout of the current screen (see Figure 16-8). Press CTRL–PRINT to print the entire buffer from the frozen screen. An explanation of Program Trace can be found in Section 6.

```
*EMDCE/LINE* BLK=00000 12/02/89  13:01  Page : 1
ASCII/8/NONE/BOP
Program Trace
Layer 3   Test fips_module_a20   state begin_init
                        restart_req sent
                        restart_con recvd
Layer 3   Test fips_module_a20   state call
                        incoming call sent
Layer 3   Test fips_module_a20   state p3
                        call_accept recvd
Layer 3   Test fips_module_a20   state bad_PR
                        invalid P(R) sent
Layer 3   Test fips_module_a20   state complete_init
                        reset_req recvd
Layer 3   Test fips_module_a20   state test_06
                        reset_c too long sent
Layer 3   Test fips_module_a20   state pass_fail
                        timeout:  no reset_req
                        ** test failed **
```

Figure 16-8 Program Trace printout in Freeze mode.

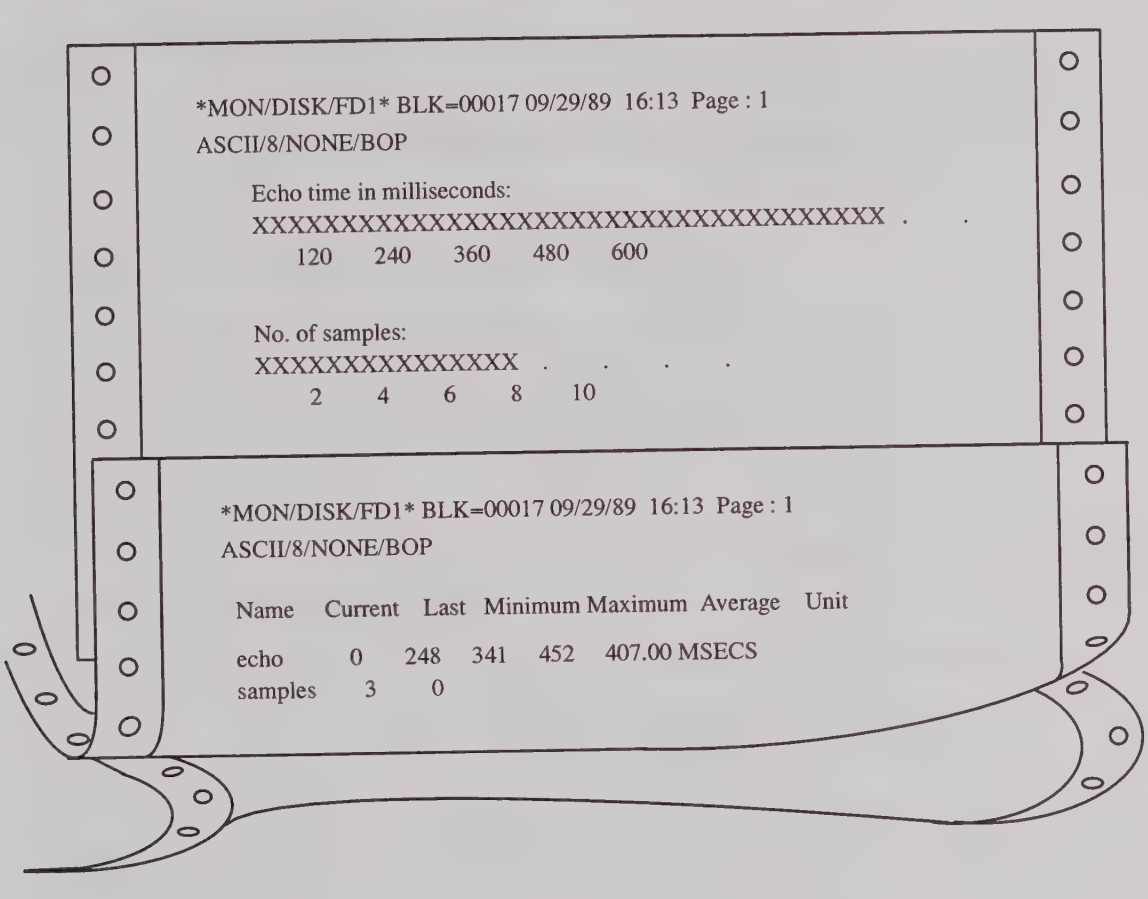
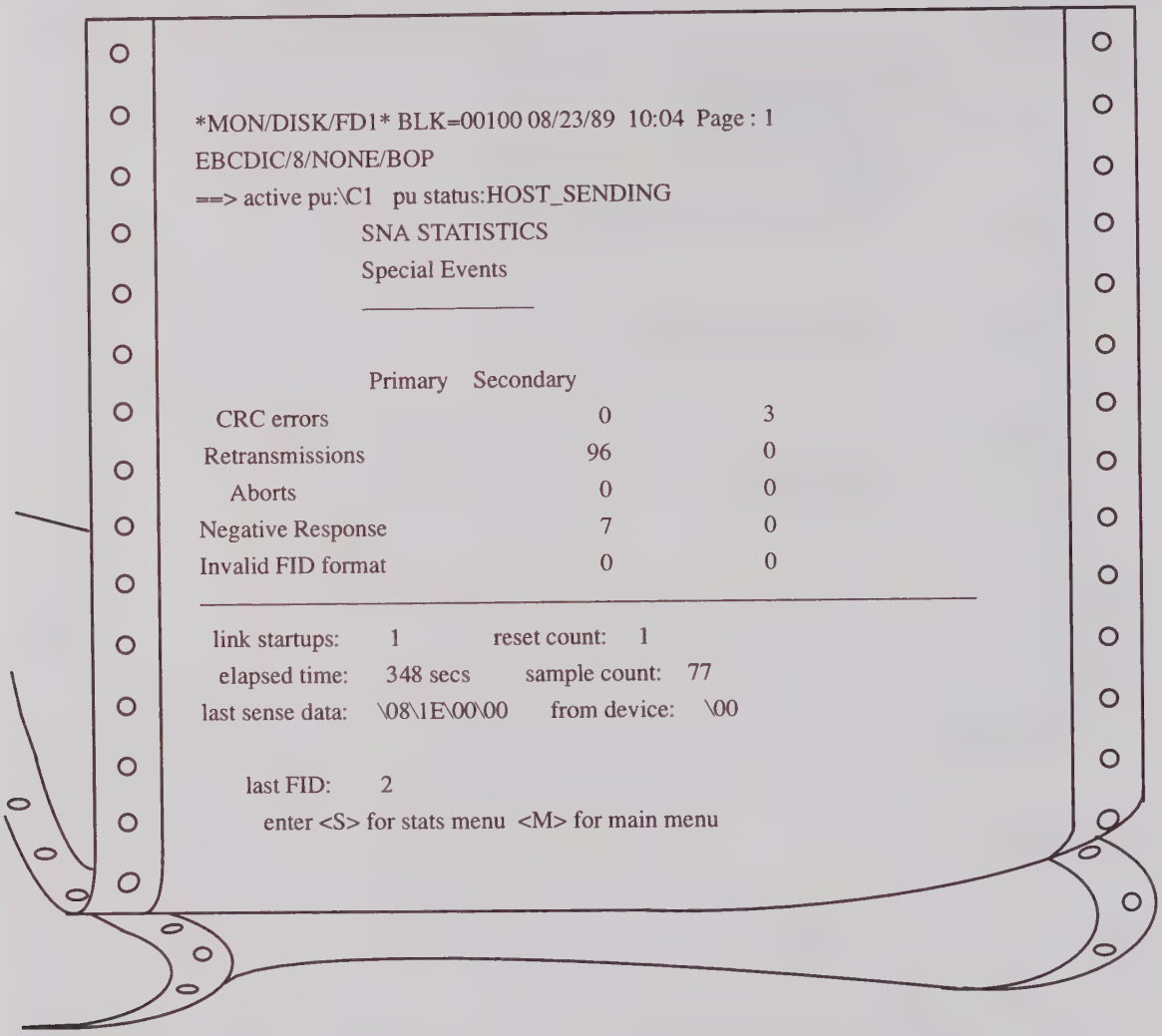


Figure 16-9 Printout of graphical (top) and tabular (bottom) statistics in Freeze mode.

(D) Printing Statistics

Tabular and graphical statistics can be accessed via softkey in Program mode and Run mode (frozen and real−time). Either of the statistics screens also may be accessed via the RUN key if STATISTICS is the **Display Mode** selected on the Display Setup menu.

Once a statistics screen is displayed, press PRINT to send the current statistics to an attached printer. Figure 16-9 is an example of graphical and tabular statistical printouts.



```
*MON/DISK/FD1* BLK=00100 08/23/89  10:04  Page : 1
EBCDIC/8/NONE/BOP
==> active pu:\C1   pu status:HOST_SENDING
                  SNA STATISTICS
                  Special Events
                  ______________


                   Primary    Secondary
  CRC errors                 0            3
 Retransmissions             96           0
    Aborts                   0            0
 Negative Response           7            0
 Invalid FID format          0            0
 ____________________________________________

  link startups:     1          reset count:    1
   elapsed time:     348 secs   sample count:   77
 last sense data:    \08\1E\00\00    from device:    \00

      last FID:      2
        enter <S> for stats menu  <M> for main menu
```

Figure 16-10 Printout of a screen from the *SNA Statistics* applications program (OPT–951–19–1) from the Display Window.

(E) Printing from Display Window

Customized screens, such as those in applications programs, can be printed in Freeze mode from the Display Window. Press PRINT to generate printouts like that in Figure 16-10.

(F) Printing User Traces

User traces can be printed in Freeze mode. Video enhancements are not indicated on the output. Press PRINT to generate a printout of the current screen. Press CTRL – PRINT to print the entire buffer from the frozen screen. User traces are explained in Section 6.

16.6 Spreadsheet Control of Printing

The Protocol Spreadsheet allows PRINT as a spreadsheet action. This causes the INTERVIEW to respond to a predetermined condition and print a line of tabular statistics for an accumulator, counter, or timer; or print a user–prompt that is sent to the printer after it has been written to the second line of the screen. (See Section 31 for an explanation of spreadsheet–controlled printing.)

The library of C routines includes several print functions. See Section 63.

16.7 Printing Disk Files

Files stored on disk can be printed from the File Maintenance menu. See Section 15.3.

17 The Time–of–Day Clock

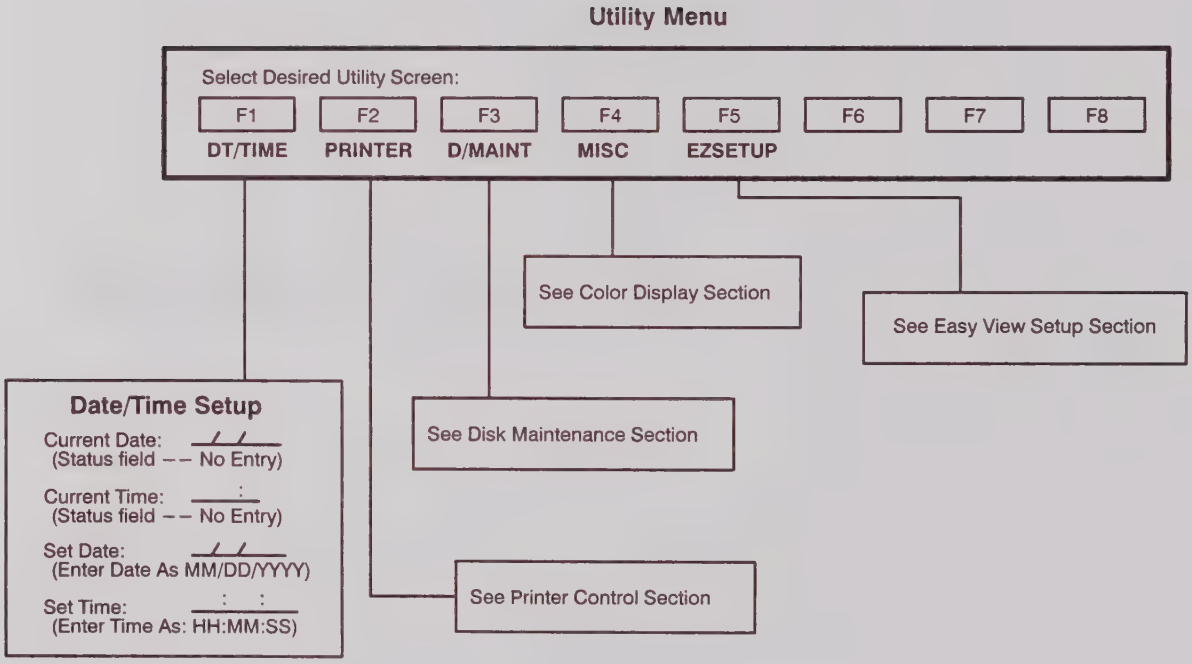


Figure 17-1 The Date/Time Setup menu.

17 The Time–of–Day Clock

The INTERVIEW has a battery–operated real–time clock. The day, month, and year as well as the current time, posted to the second, are automatically accessible as soon as you power on the INTERVIEW. Used to time/date stamp recorded data, it is also available as a Trigger Condition on the Protocol Spreadsheet. The time and date can be reset on the Date/Time Setup screen, which is a submenu of the Utilities menu.

A sample menu is shown in Figure 17-2. The current date and time appear at the top of the screen. The time is updated on the screen every second. The clock retains time and date even when powered off and adjusts itself to accommodate the length of each month. It also takes leap years into account automatically.

To reset the date, select the **Set Date** field and enter month, day, and year in that order in the spaces provided (**MM/DD/YYYY**). Use RETURN, →, or DONE to move into each field. The month and day entries may be either one or two digits; it is not necessary to enter a leading zero. If only one digit appears in any of the two–digit fields, a leading zero is assumed. The year must contain all four digits (*e.g.*, 1 9 9 7, not 9 7)

You may fill in the month, day, or year fields individually, if you prefer. (For example, if your only entry is 1_/ in the month field, only the month will be altered—it will be set to 01/—when you press XEQ.)

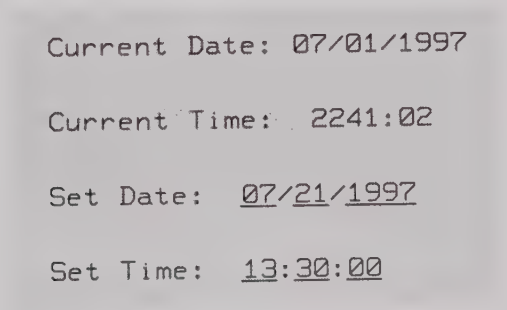


Figure 17-2 The real–time clock in the INTERVIEW is controlled on the Date/Time Setup screen.

In the **Set Time** field, hour, minute, and second entries are made as **HH:MM:SS** in a 24–hour format. You may set the hour, minutes, and seconds fields simultaneously or individually. As with the date field, a single–digit entry is assumed to have a leading zero. Figure 17-2 shows the menu filled in to change the date and time to July 21, 1997 and 1:30 p.m.

To set the clock, press the [XEQ] key when you have entered the correct date or time. Once the clock is reset, the status fields **Current Date** and **Current Time** should match your entries.

> **NOTE:** If your Ethernet–ready INTERVIEW unit is utilizing NTP synchronization, the time and date are resynchronized to the Network Time Protocol server continuously. Therefore, no changes to these fields should be necessary. In this case, changes you make via these fields will be be overwritten at the first resynchronization interval following your modifications.
>
> If you are using NTP synchronization and the *Current Time* value is off by whole–hour increments, verify that the value entered in the *UTC Offset* field on the Remote Setup Screen is accurate; see Section 2.5(J)15. Other time differences could indicate a problem with either the NTP server to which you are referencing or with the INTERVIEW unit itself.

18 Color Display

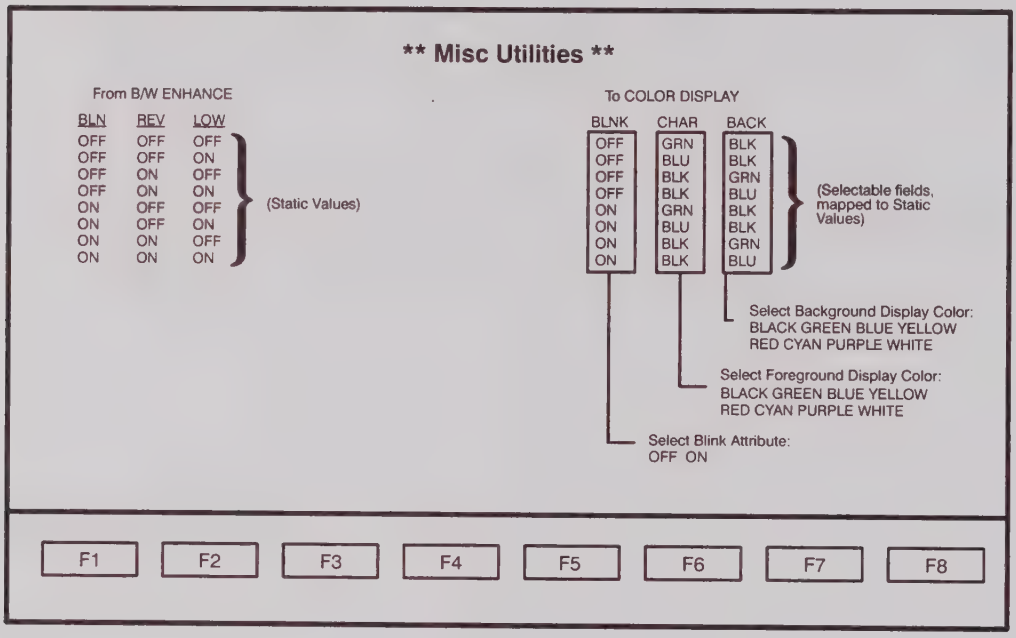


Figure 18-1 The Miscellaneous Utilities menu.

18 Color Display

Both black and white and color monitors can be used as external displays. Separate connectors are provided for each type of monitor on the rear panel of the INTERVIEW. For both monochrome and color displays, a set of video enhancements can be controlled from the INTERVIEW. For color displays, the enhancements are mapped on the Miscellaneous Utilities screen. Available options on this screen are shown in Figure 18-1.

18.1 Connectors for External Monitors

Connect a monochrome (black and white) monitor to the RS−170 connector. This is the round connector located to the far right of the rear panel (see Figure 18-2).

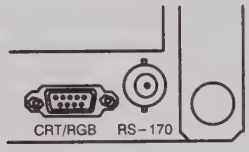


Figure 18-2 Video−out connectors for RS−170 (black and white) and RGB (color) signals are located on the rear panel of the INTERVIEW.

Connect an 8 or 16−color monitor to the 9−pin RGB connector at the right of the rear panel (see Figure 18-2). Pin configurations for the RGB connector are shown in Appendix E.

18.2 Color Control from the Miscellaneous Utilities Screen

Three of the data enhancements available as trigger actions (Blink, Reverse, and Low; see Figure 18-3) are applied differently when a color monitor is attached. There are eight possible on/off combinations for these three trigger enhancements. The user defines these eight combinations on the Miscellaneous Utilities screen.

Figure 18-3 The settings in the Trigger Enhance field (0 = off; 1 = on) are mapped to color and blink definitions selected on the Miscellaneous Utilities screen. The Hex setting does not pertain to screen colors.

The color setup portion of the Miscellaneous Utilities screen is shown in Figure 18-4. The screen is divided into two areas: the enhance combinations on the left–hand side of the screen; and the selectable display options on the right. User entries are made only on the right–hand portion of the screen.

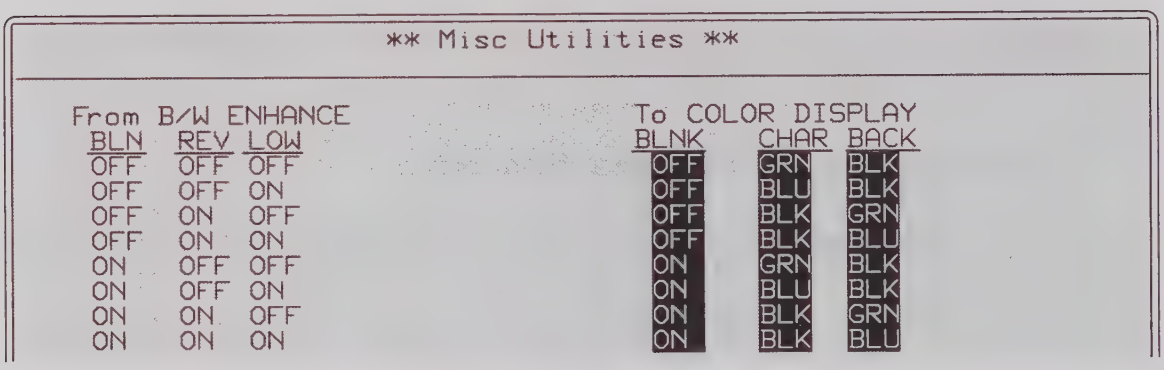


Figure 18-4 Trigger settings on the left are mapped to user–defined blink and color characteristics, selected on the right.

(A) Black and White Enhancement List

The eight enhancement combinations are labeled "From B/W ENHANCE." Each row of three settings (ON/OFF for Blink, Reverse, and Low) represents one configuration of a trigger **Enhance** field.

(B) Selectable Color Display Options

The user selects blink, foreground color, and background color options on the right half of the screen, labeled "To COLOR DISPLAY." Use the cursor keys or the RETURN key to move from field to field. The ROLL FWD and ROLL BACK keys rotate among the various selections within a field. The options which you define are matched to the static combinations to the left in the same row. Thus, when you create a trigger action **Enhance:** DTE Rev: 0 Blnk: 0 Low: 0, you are specifying that displayed DTE data take on the color characteristics defined in the first row of the Miscellaneous Utilities screen. (The Hex enhancement on a trigger does not influence color options.)

1. *Blink.* The first column under COLOR DISPLAY, labeled BLNK, determines whether blink is to be turned on or off in response to the trigger entry on the left.

2. *Character.* The center column, labeled CHAR, determines what color is assigned to the screen foreground (that is, the characters themselves) in response to the trigger **Enhance** entry on the far left. Possible foreground colors are black, green, blue, yellow, red, cyan, purple, and white.

3. *Background color.* The right-hand column, BACK, defines screen background color (that is, the square background around subsequent individual characters) in response to the trigger entry on the far left. Color possibilities are the same as for foreground.

> **NOTE:** Never use the same foreground and background color on the display if you have an eight-color monitor. (Data will not be visible.) If you have a 16-color monitor, you may be able to choose the same color in the background and the foreground and retain a contrast between data and the surrounding screen. Check the documentation for your monitor.

(C) Trigger Control of Color

When you enter the three-bit enhancement option as an **Enhance** action on a Trigger Setup screen (see Figure 18-3) or as a Layer 1 ENHANCE action on the Protocol Spreadsheet, you are designating the color scheme which the screen will adopt whenever that trigger becomes true. More specifically, you are indicating what color the next character becomes (foreground color), whether the character blinks, and what color the small rectangular box surrounding the character will be (background color). These three characteristics apply to subsequent characters and their background until another trigger alters the color scheme.

ENHANCE actions above Layer 1 are applied to specific lines on the protocol-trace display for the given layer. REVERSE, BLINK, and LOW selections may be configured at Layer 3, for example, so that a Call Request packet receives a different color enhancement from a Clear Request.

When you program a trigger, only the **Enhance** entries for Blnk, Rev, and Low influence screen characteristics. (The Hex field always causes characters to be displayed in hexadecimal, or turns off the hexadecimal character enhancement.) An **Enhance** entry of 1 equates to "ON"; an entry of 0 equates to "OFF."

> **NOTE:** A "Don't Care" (X) after Blnk, Rev, or Low leaves the enhancement at its previous setting, making the color which results from the trigger dependent on the effects of previous triggers.

Each time you run the program, all enhancement settings are initially reset to zero. If no trigger entries are made, or if the enhance trigger never becomes true, the color monitor retains its own color settings. (Refer to the technical documentation for the monitor to determine how colors are set internally.)

(D) Color Graphics

Several colors are offered for bar graphing on the Graphical Statistics menu. These colors are always displayed against a dark background. Their use is not related to the color setup screen. See Section 22 for a description of color graphics.

18.3 Black and White Data Enhancements

Blink and reverse data enhancements for the INTERVIEW's plasma display are available when a monochrome monitor is installed. In addition, a low intensity enhancement can be produced on a monochrome monitor. These display highlights are controlled by triggers (see Figure 18-3), either on the Protocol Spreadsheet or on Trigger Setup screens. Refer to Section 26 or 32 for a description.

19 Easy View Setup

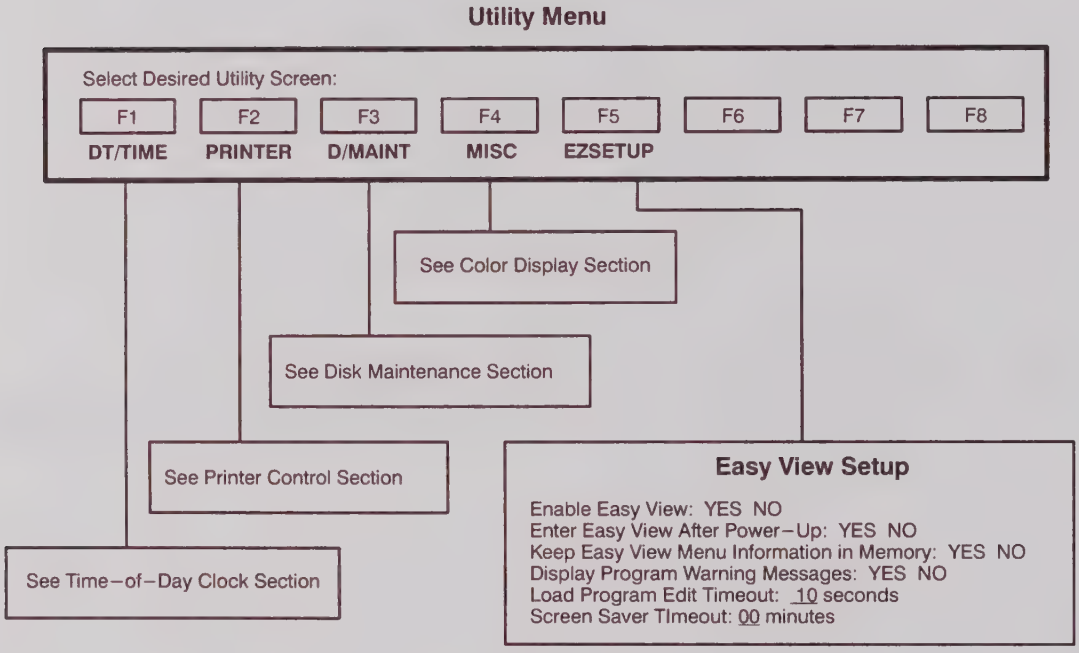


Figure 19-1 The Easy View Setup menu.

19 Easy View Setup

There are five parameters that control Easy View operation. They are set in the Program–mode Easy View Setup screen. See Figure 19-2. To access this screen, press UTIL, EZSETUP from the main Program menu.

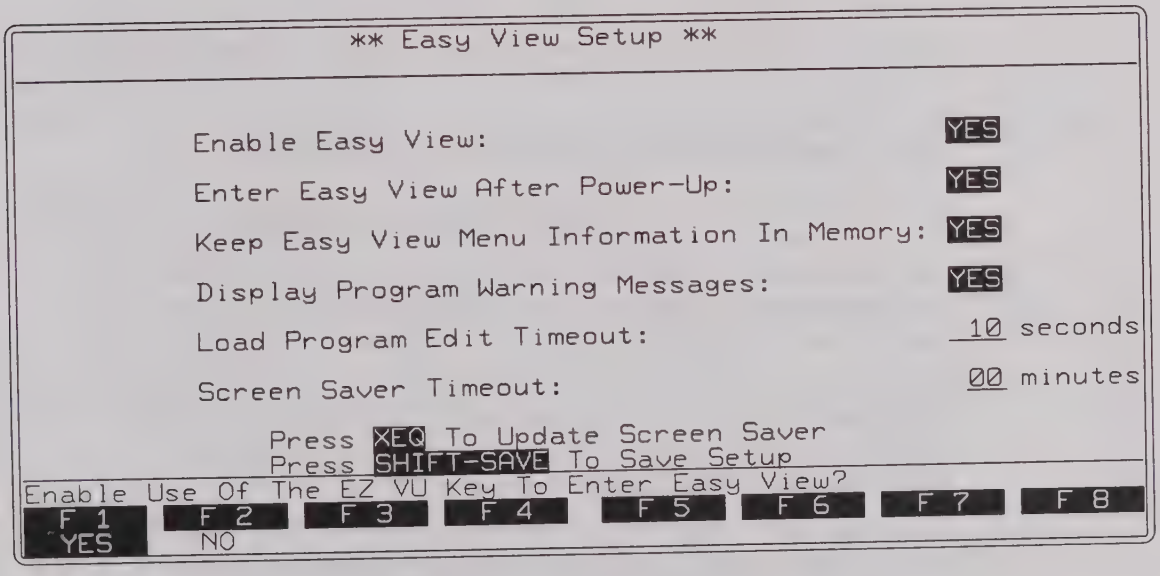


Figure 19-2 The Easy View Setup screen.

The default selection of each parameter on the Easy View Setup menu is YES . If you press [SHIFT]–[SAVE], the configuration of this screen is saved in a file called *ezview_setup* in the */sys* directory of the boot–up disk. Once saved, the parameters are automatically loaded during boot–up.

19.1 Enable Easy View

The first parameter determines whether or not the [EZ VU] key is enabled. The default selection is YES . Once enabled, [EZ VU] moves the user back and forth between Program–mode screens and Easy View menus. (See Section 4.) Setting this parameter to NO disables access to the Easy View menus. An alarm sounds if you press [EZ VU] when it is disabled.

19.2 Enter Easy View After Power–Up

The second parameter determines whether the INTERVIEW enters Program mode or Easy View following boot–up or a manual reset from the keyboard.

If you set this field to YES, there is a five–second pause at the power–up screen before you see the first Easy View menu. Use the EZ VU key to exit Easy View and access Program–mode screens.

> **NOTE:** During boot–up, the INTERVIEW checks the boot–up disk for a file named */usr/default* and the hard disk for a file named */usr/user_intrf*. (See Sections 2.1 and 2.2.) If it locates either, it does not enter Easy View following boot–up regardless of the setting of this parameter.

If you select NO, the INTERVIEW is in Program mode at power–up. Use the EZ VU key from any Program–mode screen to enter Easy View.

19.3 Keep Easy View Menu Information in Memory

The third parameter controls whether the contents of the *mstrmenu.cmp* file and the most recently accessed text file are held in memory while the INTERVIEW runs a selected program. *mstrmenu.cmp* is the Easy View menu information file that contains all of the information needed to present the menu screens and permit the user to interact with them. This compiled file resides in the */MENU* directory on the hard disk.

Easy View keeps track of your path through the menus, even after you have exited Easy View. Each time you return to Easy View, you are automatically located in the same menu from which you exited. When this field is YES, you can exit Easy View to run an application program without losing your place in the menus.

If you select NO, your position in the Easy View menus is lost whenever you load a program or layer package. When you return to Easy View, the first menu in the hierarchy is displayed. It is possible that a program you select to run from Easy View may be too large to compile with *mstrmenu.cmp* and a text file in memory. In this case, set this parameter to NO.

19.4 Display Program Warning Messages

By default, the warning message shown in Figure 19-3 is displayed whenever you select a program to run from an Easy View menu. The message is displayed before the program is loaded and indicates that loading the program will result in the current setup parameters, triggers, and spreadsheet being overwritten.

If you do not want this message presented, disable it on the Easy View Setup screen. You may also disable the warning message temporarily (until the INTERVIEW is turned off) by pressing D when the message is first presented.

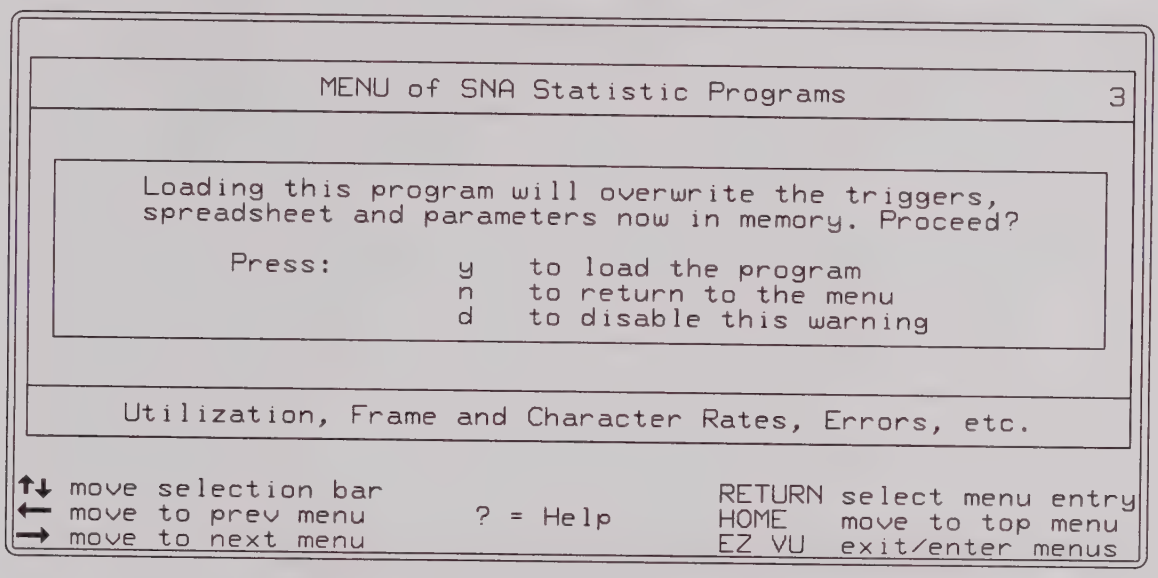


Figure 19-3 A warning message displayed prior to loading a program.

19.5 Load Program Edit Timeout

In this field, specify a length of time that the INTERVIEW should provide for editing Easy View setups. The default timeout is 10 seconds. Any value in the range 1–999 is valid as a timeout. If you leave the field blank, or enter a zero, the program will run, bypassing the editing screen.

This feature is particularly useful during remote operation. By entering a timeout, you can ensure that the remote unit will not enter Run mode before you have a chance to edit program parameters.

19.6 Screen Saver Timeout

The INTERVIEW's Screen Saver Timeout feature allows the user to specify the number of minutes the screen display remains visible when the keyboard is not in use. The screen saver blanks the internal EL display as well as any optional direct external displays of the unit, such as an external color monitor and an external RS170 (monochrome monitor); it also disables the unit's keyboard so that *any* key pressed will only restore the display. However, any PC remotely controlling the unit is unaffected by the screen saver.

The default for the **Screen Saver Timeout:** field is zero minutes, which means no timeout is requested and the screen remains displayed. Enter any integer in this field in the range 1–99 to specify the number of minutes the display remains visible before the screen goes blank. Press [XEQ] to update the Screen Saver Timeout entry; press [SHIFT]–[SAVE] to save the setup.

To return the display to the screen after the unit times out and the screen blanks, press any key as stated above. (Note that any key pressed will unblank the screen *only*; normal key usage is not in effect with a blank screen but will be restored after the screen display returns.)

20 Easy View Maintenance

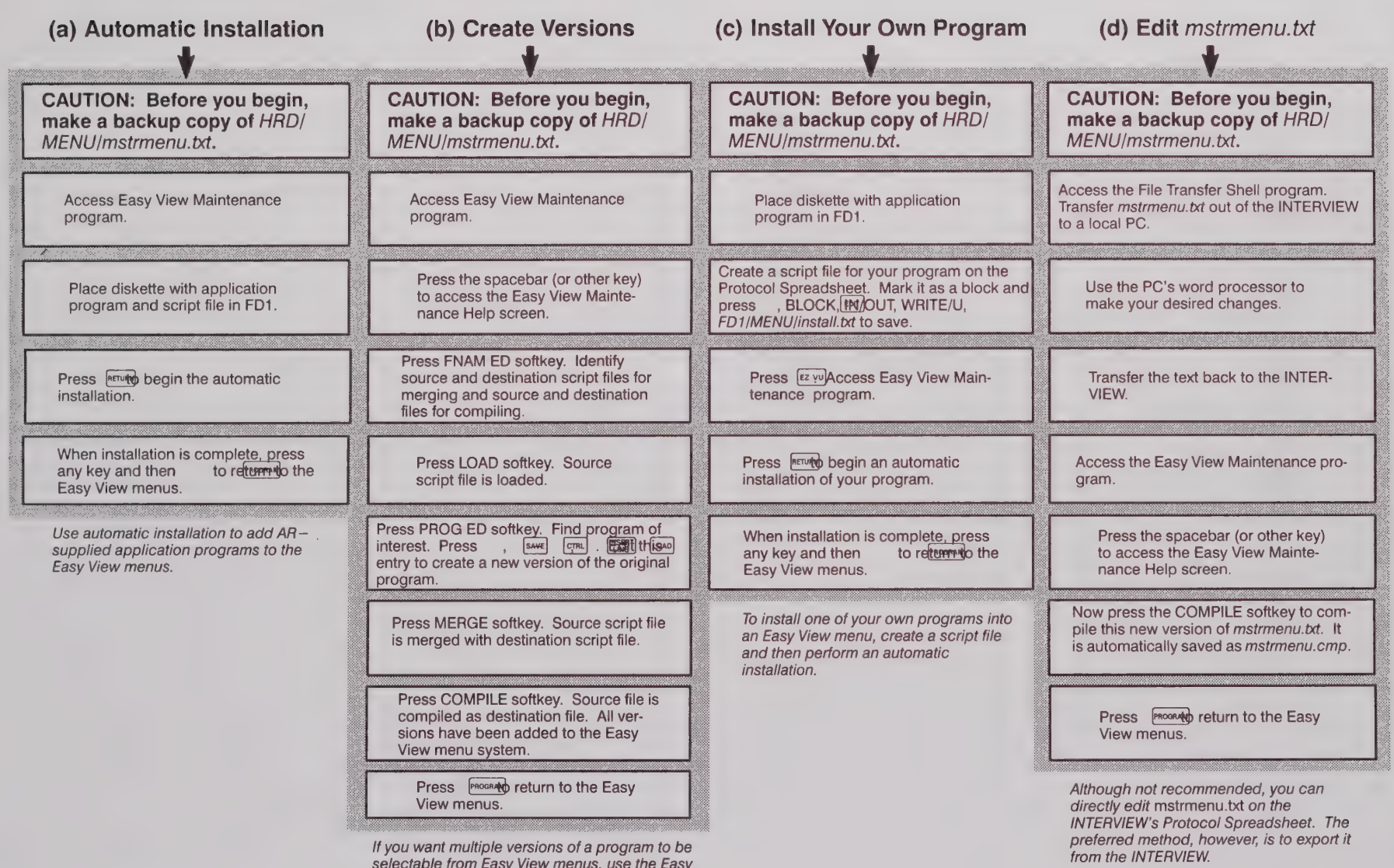


Figure 20-1 Overview of the different uses of the Easy View Maintenance program.

20 Easy View Maintenance

This section addresses the Easy View Maintenance program, one of the selections on the Utilities, Model and Demonstration Programs menu. See Figure 20-2. It also provides information on creating and editing script files used in the installation procedure.

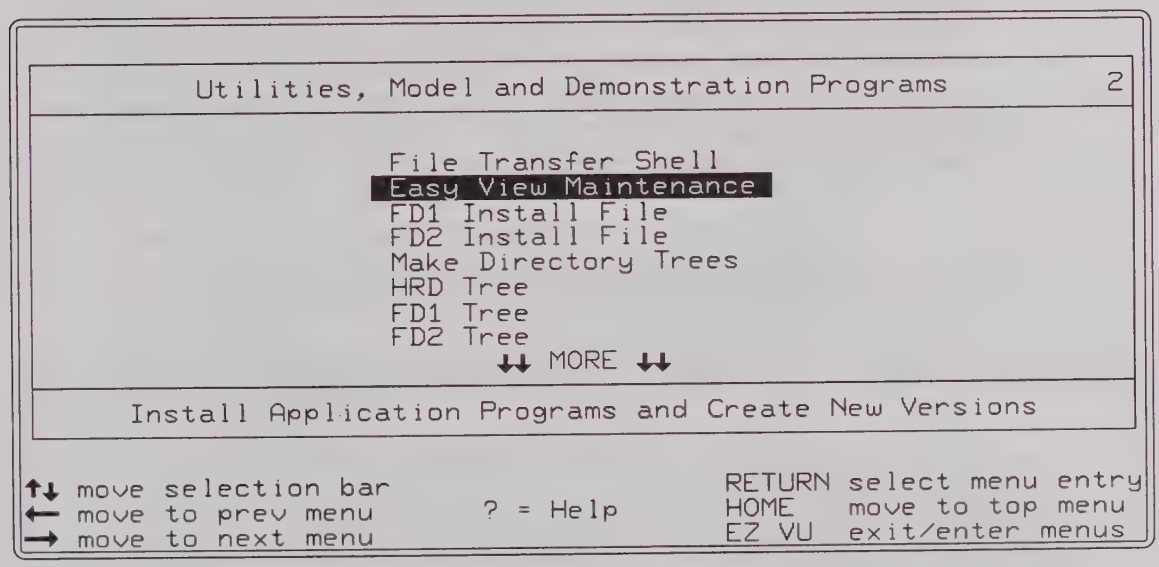


Figure 20-2 Easy View Utilities, Model and Demonstration Programs menu.

Figure 20-1 provides an overview of the different uses of the Easy View Maintenance program. Typically, you will use the Easy View Maintenance program to install AR–supplied application programs or your own programs into the Easy View system. You can also use it to create multiple versions of these application programs—for example, versions with different line speeds or line/disk source selections. Then, all versions are selectable from the Easy View menus.

The Easy View Maintenance program *adds* entries to the menus. It cannot modify, reorder, or remove existing menu entries. Whenever you add a selection to an Easy View menu, you are modifying the Easy View menu file.

20.1 The Easy View Menu File

The menu information file, *mstrmenu.cmp*, contains all of the information needed by the Easy View system to present the menu screens and permit the user to interact with them. This compiled file must reside in the */MENU* directory on the hard disk.

The source text file (also referenced as the script file) for *mstrmenu.cmp* is an ASCII file named */MENU/mstrmenu.txt*. All script files, including *mstrmenu.txt*, follow the format explained in Section 20.5.

When you perform an automatic installation of an application program (or customized versions of an application program) via the Easy View Maintenance utility, the INTERVIEW incorporates script information about the new program(s) into *mstrmenu.txt* and then generates a new version of *mstrmenu.cmp*.

20.2 Accessing the Maintenance Program

Regardless of which function you want the Easy View Maintenance program to perform, first go to the Utilities, Model and Demonstration Programs menu—from the Easy View Main menu, position the selection bar over the UTILITIES menu item and press RETURN. Now position the bar over Easy View Maintenance, as shown in Figure 20-2. Press RETURN again. You may be questioned about whether or not you want to load the program. If so, press Y. Then, the screen shown in Figure 20-3 should appear.

> **NOTE:** The instructions in this section assume that you are currently in the Easy View system. See Section 4 on Easy View general operation and Section 19 for information on enabling Easy View.

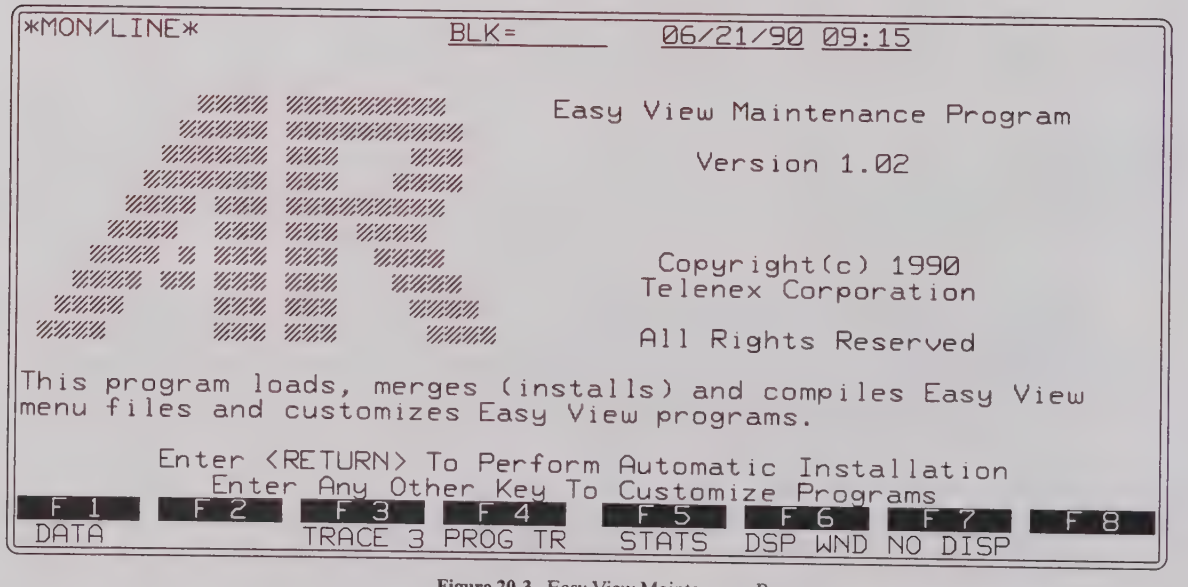


Figure 20-3 Easy View Maintenance Program.

This screen identifies the Easy View Maintenance program (and its version number) and gives a general description of its scope and purpose. Decide which function you want the program to perform.

CAUTION: Before you begin, we strongly recommend that you make a backup copy of mstrmenu.txt.

To begin an automatic installation, press RETURN from this screen. See Section 20.3.

Press a key other than RETURN to access the screens for customizing programs. See Section 20.4.

20.3 Automatic Installation

A library of selected application programs has been factory–installed in the *AR_APPS* directory of the hard disk. The subdirectories under *AR_APPS* categorize these programs by protocol and function. (See Section 20.7 for more information on the directory structure.) Additional AR–supplied application programs, available as options, may be added to the Easy View menus via the Easy View Maintenance utility.

If you do not need customized versions of a new application program and want to perform an automatic installation, place the diskette containing the application program and its script file into FD1, the left–hand floppy disk drive. (The script file that accompanies all AR–supplied application programs is named */MENU/install.txt* and resides on the same program diskette.) Press RETURN from the Easy View Maintenance program start–up screen (Figure 20-3).

The INTERVIEW automatically installs the application program into the appropriate Easy View menu so that it may be loaded, compiled, and run from Easy View. The INTERVIEW looks to FD1 for the program's script file and then merges it into *mstrmenu.txt* on the hard disk. Finally, the revised *mstrmenu.txt* is compiled to create an updated version of *mstrmenu.cmp*.

If you press PROGRAM to abort the process and exit the program, you are returned to the Utilities, Model and Demonstration Programs menu. Once the installation is complete, press any other key. The Easy View Maintenance Help screen is presented. See Figure 20-13. To return to the Easy View menus, press PROGRAM.

You may also use the automatic installation feature to add selections to the Easy View menus for your own programs. First, however, create the script file the Easy View Maintenance program needs to perform an installation and save it to FD1 as */MENU/install.txt* (the default name for the source script file during automatic installation). See Section 20.5.

20.4 Customizing Programs

Refer again to Figure 20-3. Notice that if you press a key other than RETURN, you can customize the setups of application programs. Use this feature to create multiple versions of a program.

An overview of the screens and commands used in creating program versions is presented in the example below. Detailed explanations for screens and commands are presented in the sections which follow.

(A) Example

Before we begin, make a backup copy of *mstrmenu.txt*.

For this example, we will load the script file *mstrmenu.txt* and create a new version of the Easy View ASYNC monitor program, Standard ASYNC Setup.

Access the Easy View Maintenance program. Then press the spacebar, or some other key except PROGRAM, to present the Easy View Maintenance Help menu. See Figure 20-4.

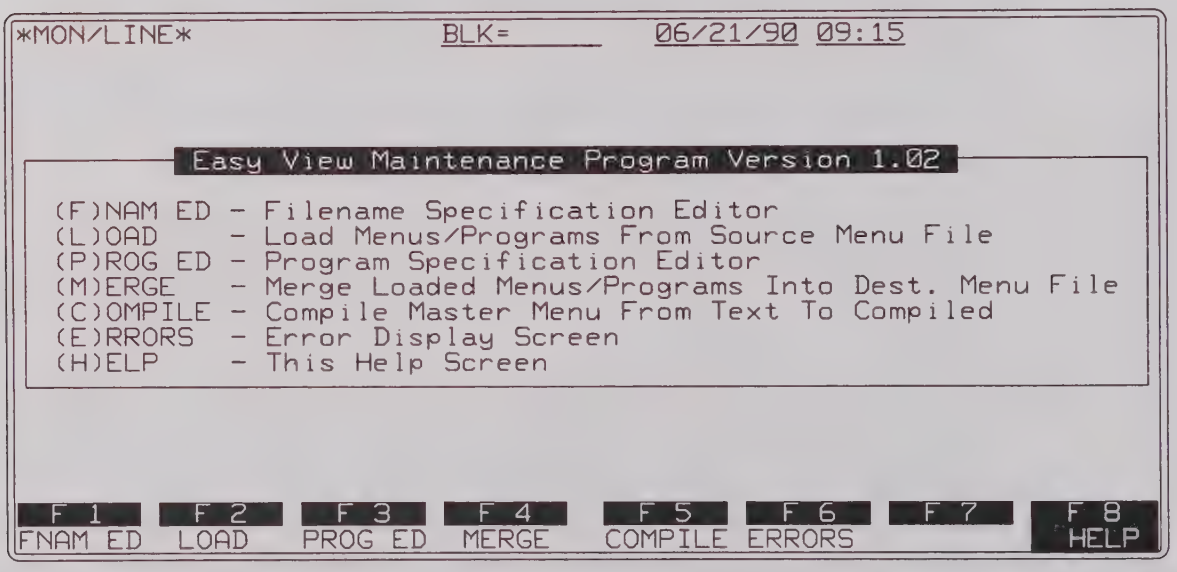


Figure 20-4 Easy View Maintenance Help menu.

This screen provides a brief description of the available screens and commands.

First, we need to indicate the script file we want loaded. The program we want to create new versions for is in this file. Identify the file on the Filename Specification Editor screen. Press FNAM ED. Type over the existing entry to change the name of the **Source (Load From) Menu File Pathname** to *HRD/MENU/mstrmenu.txt*. See Figure 20-5. For now, leave the other entries intact.

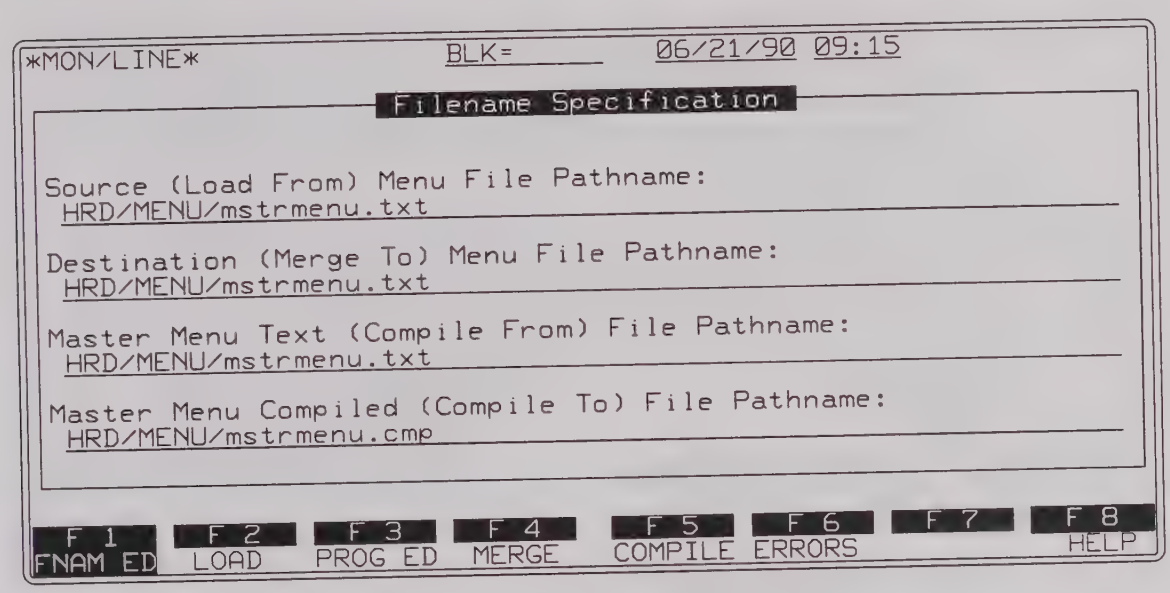


Figure 20-5 Load *mstrmenu.txt* for editing.

Now press LOAD. All of the setups for all of the programs you can access from Easy View menus are loaded from *mstrmenu.txt*.

Press PROG ED. The screen shown in Figure 20-6 should appear.

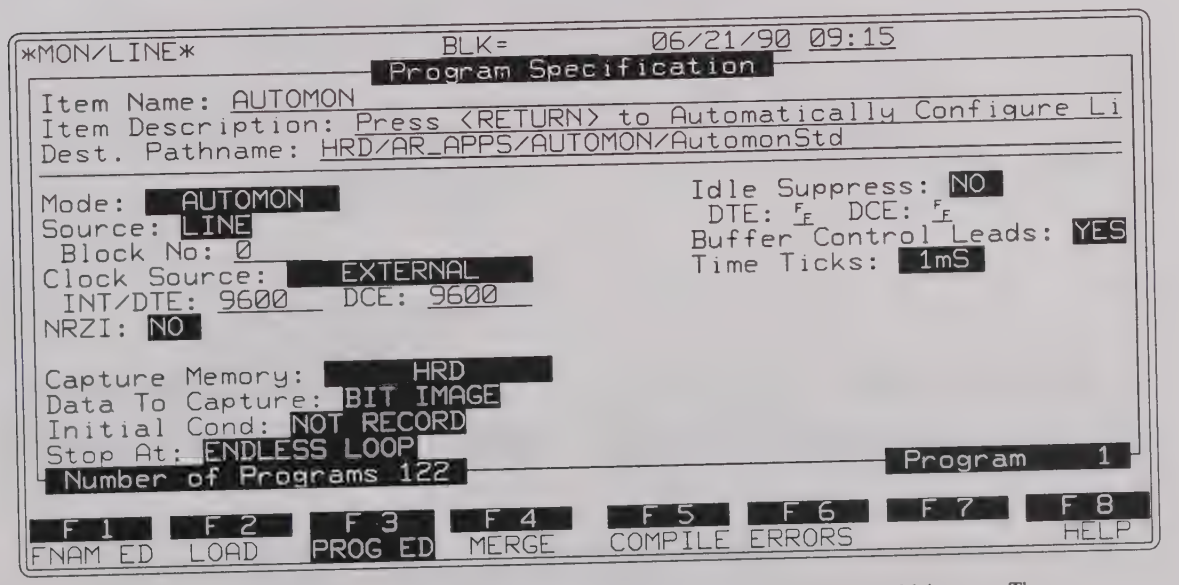


Figure 20-6 This screen shows the setup for the *AUTOMON* selection on the Easy View Main menu. The setups for all Easy View programs are contained in *mstrmenu.txt*.

Look at the **Item Name** field at the top of the screen. The names in this field match selections on the Easy View menus. Use the NEXT PAGE key to scroll through the listings from *mstrmenu.txt*. Watch the **Program** number in the lower right–hand corner of the screen. When it reaches 18, you should find the entry for Standard ASYNC Setup. It is illustrated in Figure 20-7.

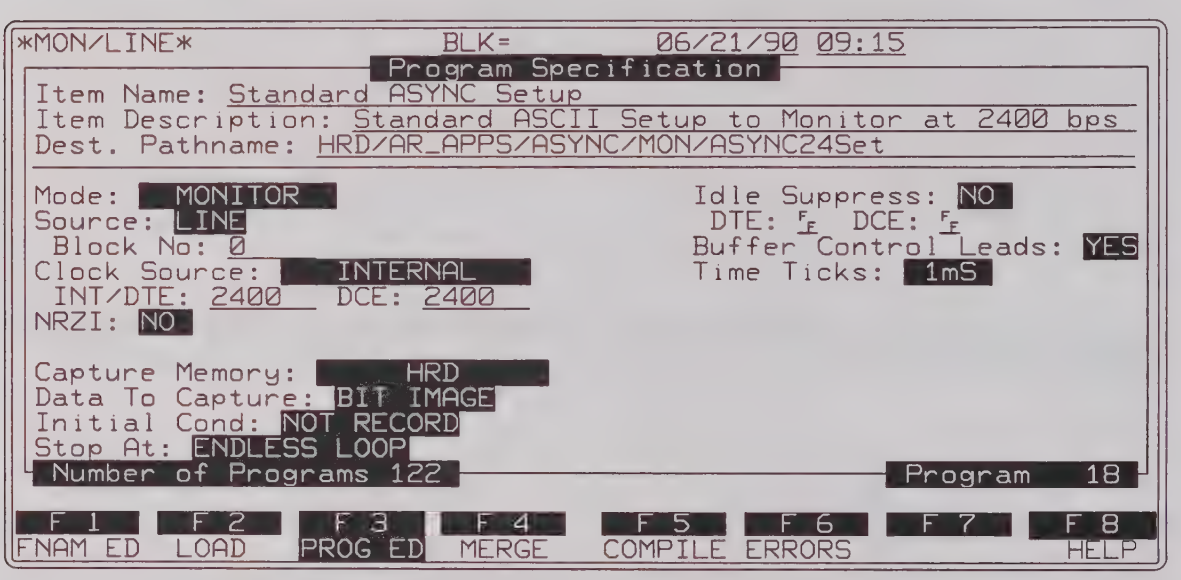


Figure 20-7 This screen shows the setup for the *Standard ASYNC Setup* selection on the Menu of ASYNC Monitor Programs. This setup is part of *mstrmenu.txt*.

Press SAVE to keep this entry accessible in the program listings.

Now press CTRL – INSERT LINE. The dummy entry shown below is inserted into the listing. Do not directly edit this entry. Dummy entries are place holders only.

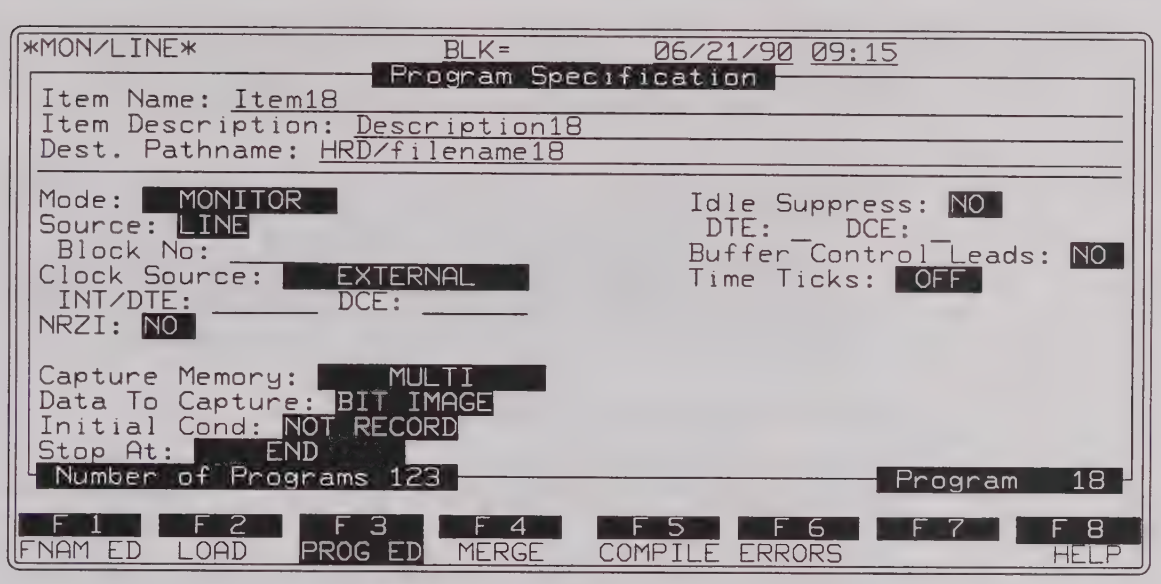


Figure 20-8 Press CTRL – INSERT LINE to create a dummy entry.

Instead, press LOAD. The dummy entry is overwritten by a numbered version of the (Standard ASYNC Setup) program you just saved. See Figure 20-9.

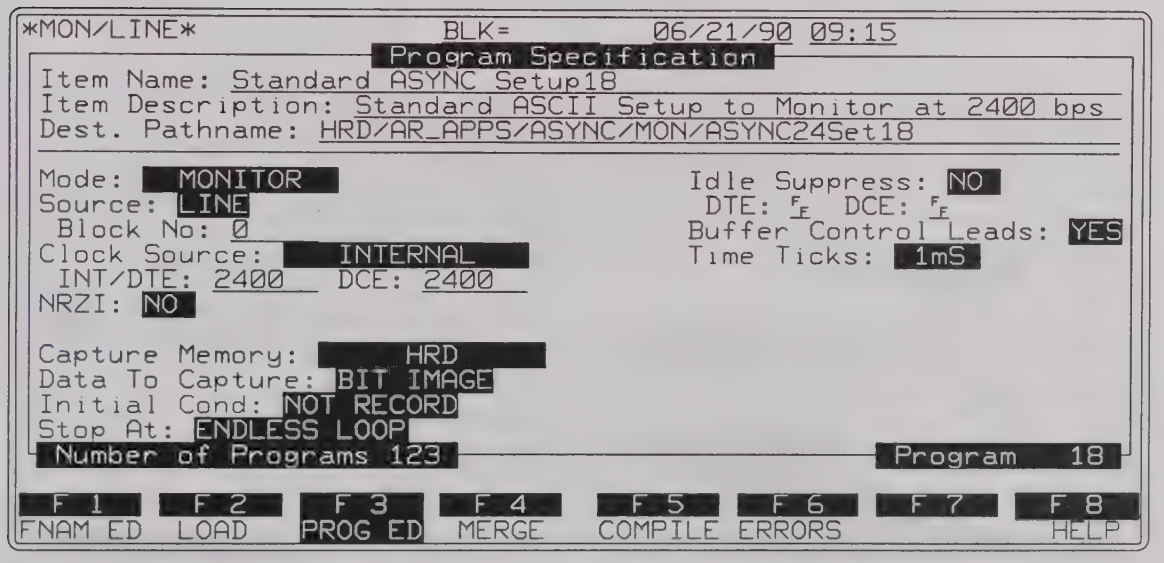


Figure 20-9 Numbered version of an Easy View program. Edit this entry to create your own version.

This is the entry we will edit to create a new version of this program. (The original is still in place; press NEXT PAGE to view it. Then return to the numbered version by pressing PREV PAGE.)

For this example, we will create a version of the Standard ASYNC Setup program that monitors a line speed of 9600 bps. Text you type on this screen overwrites existing text. The → and ← keys move the cursor within a field. RETURN, ↓, and ↑ move the cursor between fields.

> **NOTE:** Line speed, and other selected line–setup parameters, can be changed on–line before the program is run. See Section 4.4(C). These edited Easy View parameters remain in effect until you change them again, or until the unit is powered off. If you want the changes to be permanent, use the Easy View Maintenance program as described below.

Type in a new **Item Name:** *Standard ASYNC Setup (9600).*

The **Item Description** field contains the text that appears toward the bottom of Easy View screens when the selection bar is over the menu item. Change this entry to read *Standard ASCII Setup to Monitor at 9600 bps.*

The **Dest. Pathname** field identifies the name and location of the program to run when the menu item has been selected. Give our new program the following pathname: *HRD/AR_APPS/ASYNC/MON/ASYNC96Set.*

Now we are ready to modify the setup for the program. The only change we want to make is to the line speed for internal clock. Use the RETURN key to move to the **INT/DTE:** field. Change the speed from 2400 to 9600.

The information for our new version is complete. See Figure 20-10.

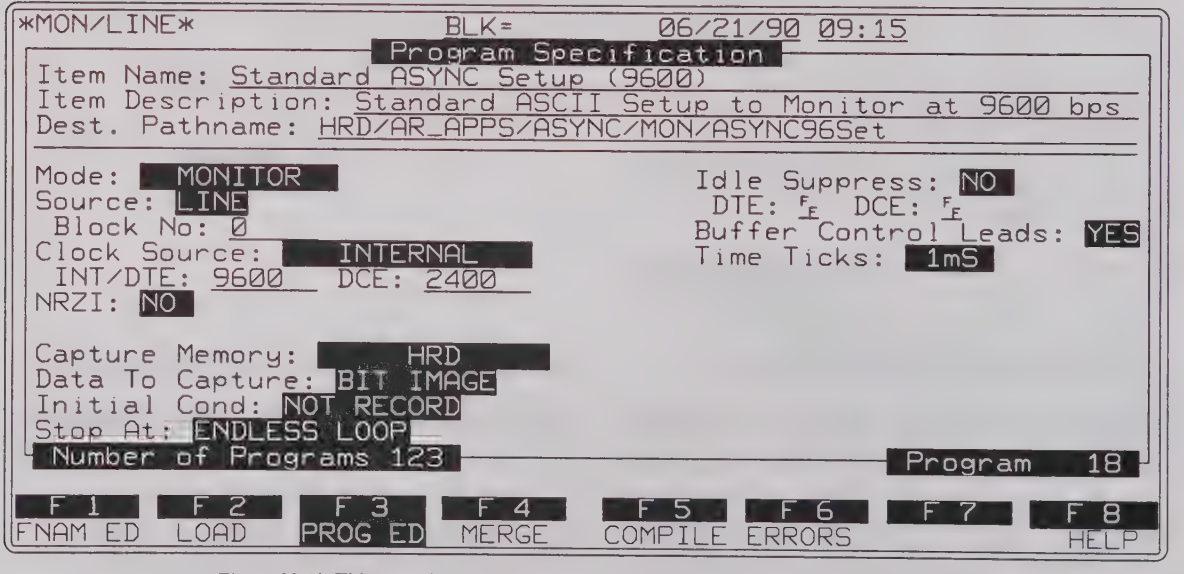


Figure 20-10 This setup for a new version of the Standard ASYNC Setup program will be added to *mstrmenu.txt* when the MERGE command is executed.

To incorporate this information into the existing *mstrmenu.txt* file, press the MERGE softkey. There will be an indication of errors at the end of the merge operation. Press the ERRORS softkey to view error messages. See Figure 20-11.

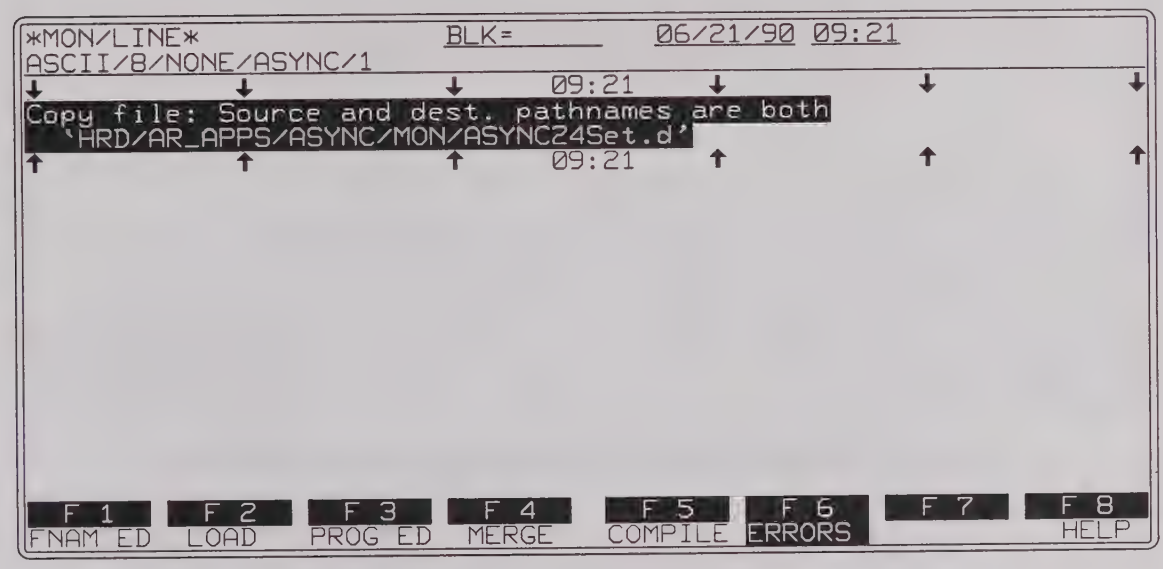


Figure 20-11 The Error Display screen.

The message you see indicates that an attempt was made to copy the help file that exists for the original Standard ASYNC Setup program. Duplicate copies of help files by the same name cannot be made. You can disregard this message, however, since the help file for the original will still be accessible and generally applicable for our new menu item.

Now press the COMPILE softkey. This command recreates the menu information file.

Press PROGRAM to return to the Easy View menus. If not already at the top of the menu hierarchy, press HOME to return to the first location on the Easy View Main menu. Use the ↓ key to move the selection bar to ASYNC and press RETURN. Select the MONITOR item on the Menu of ASYNC Program Types, Tutorial(s) & How To's. Press RETURN. Our new version of the Standard ASYNC Setup program appears at the end of the Menu of ASYNC Monitor Programs. See Figure 20-12.

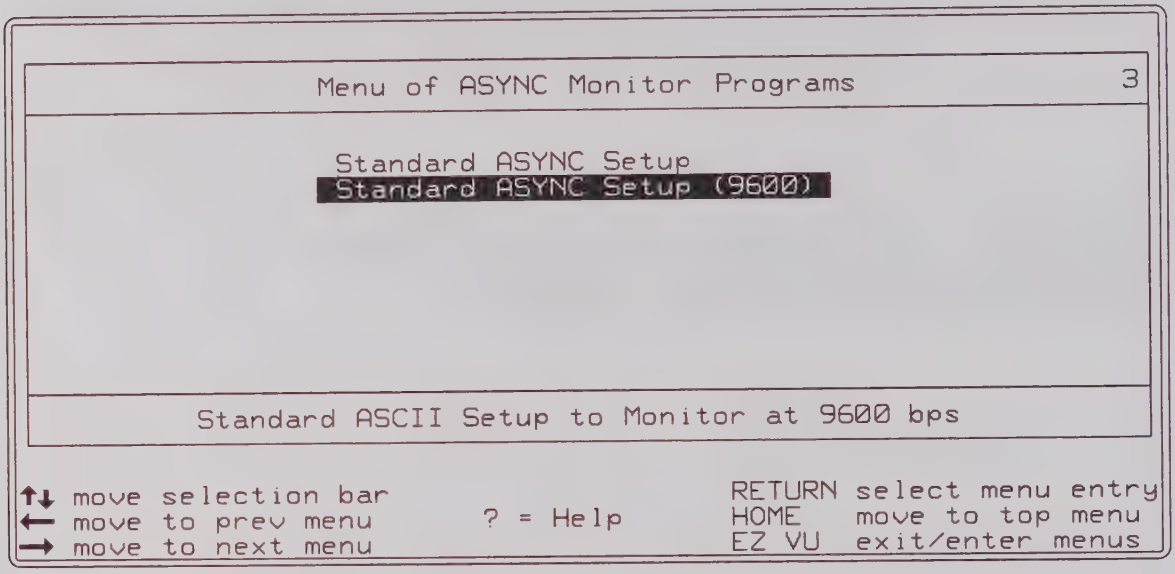


Figure 20-12 Our new entry, *Standard ASYNC Setup (9600)*, appears at the end of the Menu of ASYNC Monitor programs.

Now that you have a general idea of how to create a new version of an Easy View program, let's examine each screen and command in more detail.

(B) Easy View Maintenance Help Menu

The screen shown in Figure 20-13 is the Easy View Maintenance Help menu. It provides a brief description of each softkey function. Pressing the function key (or the first letter of the softkey label) either accesses another screen or performs a command. LOAD, MERGE, and COMPILE are commands.

The softkeys are ordered according to the sequence of tasks you will perform. To customize an application program, first identify the script filename on the Filename Specification screen. Next, load into memory the setups from the program(s) referenced in the script file via the LOAD command. Then, move to the Program Specification Editor screen. Once you find the program of interest, make the appropriate changes to the setups and menu information to create new versions. MERGE the new information into the master menu script file. As a final step, COMPILE the master menu information file. Refer again to Figure 20-1.

NOTE: Except for program editing, the automatic installation performs these same operations using the files named on the Filename Specification screen.

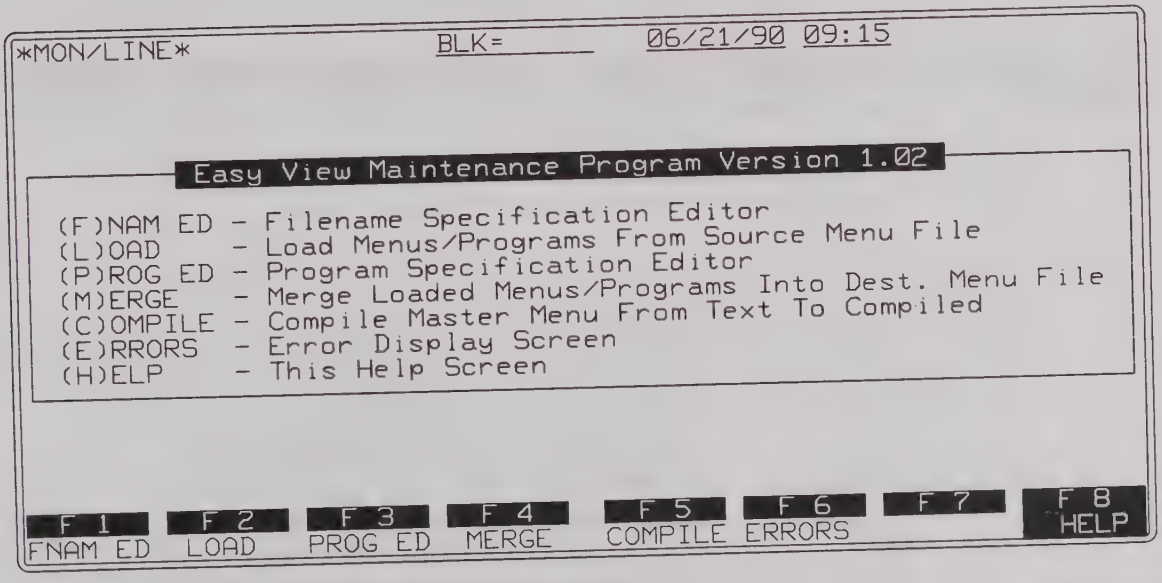


Figure 20-13 Help screen for Easy View Maintenance program.

(C) Filename Specification Screen

The source and destination files for each of the commands are identified on the Filename Specification screen, shown in Figure 20-14. Before you can customize a program, you need to identify the appropriate files. Press F or the FNAM ED softkey to access this screen.

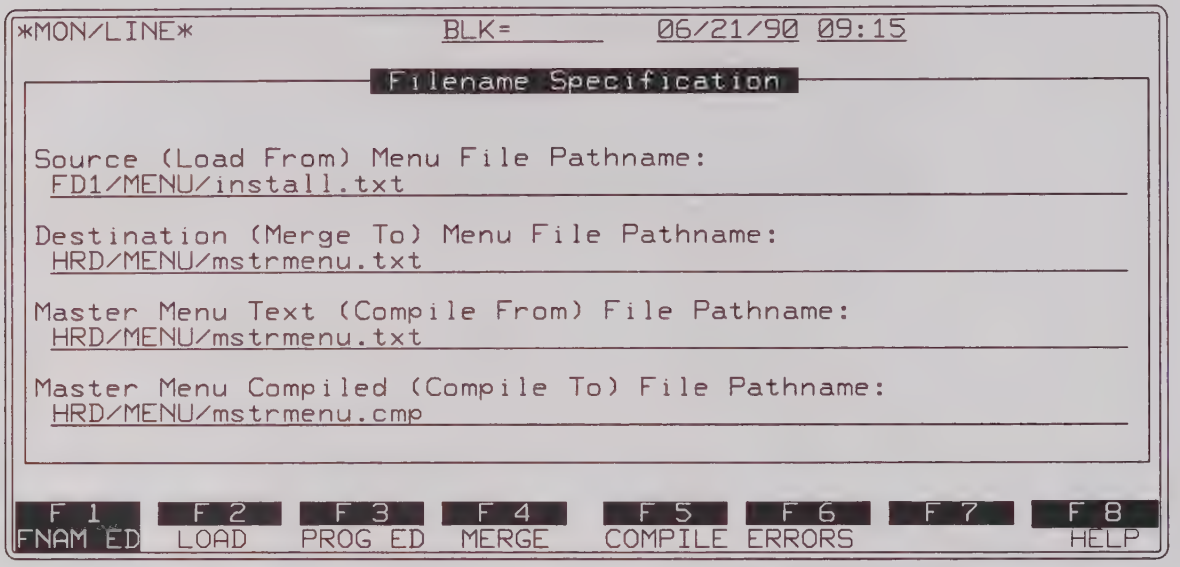


Figure 20-14 Filename Specification Editor screen.

The following editing keys function in the fields on this screen:

INSERT CHAR	Inserts a character at the cursor position.
RUB OUT	Deletes the character immediately preceding the cursor position.
DELETE CHAR	Deletes the character at the cursor position.
DELETE LINE	Deletes the line containing the cursor.
CLEAR	Clears the field containing the cursor.
SHIFT – CLEAR	Clears the field from the cursor position to the end.
SHIFT – →	Moves the cursor to the end of the current field.
SHIFT – ←	Moves the cursor to the beginning of the current field.
→	Moves the cursor forward one character, or to the next field.
←	Moves the cursor back one character, or to the previous field.
TAB or RETURN	Moves the cursor to the next field.
SHIFT – TAB or SHIFT – RETURN	Moves the cursor to the previous field.
↓	Moves the cursor to the same location in the field below.
↑	Moves the cursor to the same location in the field above.

For each field on the screen, enter the pathname of a file, beginning with a drive specifier—*HRD*, *FD1*, or *FD2*. Filenames are case sensitive. Enter the filename exactly as it appears in the INTERVIEW file maintenance directory. If you do not include the drive specifier, an error message is displayed. You will not be able to enter a name that exceeds the maximum length of 126 characters (including the drive specifier).

1. *Source (Load From) Menu File Pathname.* The file named in this field is automatically loaded when you elect to customize programs. It is also the object of the LOAD command.

   The default pathname is *FD1/MENU/install.txt*, the name of the script file for AR–supplied application programs. The automatic installation uses the script file named in this field as the source for the install.

   If you are creating a script file to install your own program, either name your script file *FD1/MENU/install.txt* (case does make a difference) or change the entry in this field.

2. *Destination (Merge To) Menu File Pathname.* When you use the MERGE command, the **Source** file named in the field above is merged into the file named here.

   The default filename is *HRD/MENU/mstrmenu.txt*. In most instances, you do not need to change the entry in this field.

3. *Master Menu Text (Compile From) File Pathname.* This is the source file for the COMPILE command. Typically, it is the same file as the **Destination** file for the MERGE command. Again, the default filename is *HRD/MENU/mstrmenu.txt*.

4. *Master Menu Compiled (Compile To) File Pathname.* The file named in this field is the destination file for the COMPILE command. The default filename is *HRD/MENU/mstrmenu.cmp*. It is the compiled version of all the Easy View menu information.

(D) Load Command

Now press LOAD to load the setups for all the programs referenced in the **Source** file (named in the Filename Specification screen). The setups may be modified on the Program Specification screen.

A rotating arrow in the lower, right–hand corner of the display indicates that the load is in progress. Press BRK if you want to abort the LOAD operation.

> **NOTE:** The default Source file on the Filename Specification screen is automatically loaded when you elect to customize programs. If this is the file you wish to modify, you may bypass the LOAD command.

(E) Program Specification Screen

The setups of all of the programs referenced in the **Source** file (named in the Filename Specification screen) are available for editing in the Program Specification screen, shown in Figure 20-15. The versions you create on this screen are incorporated in the **Destination** file when you execute the MERGE command.

The fields in the top portion of the screen, **Item Name**, **Item Description**, and **Dest. Pathname** identify one of the programs referenced in the loaded script file. The fields in the bottom portion of the screen display the current setup parameters for the file specified in the **Dest. Pathname** field.

Use the cursor keys PREV PAGE and NEXT PAGE to scroll through the program listings. SHIFT–PREV PAGE moves you to the beginning of the list, SHIFT–NEXT PAGE to the end. Notice that the field entries on the Program Specification screen change as you scroll through the list of files.

If you scroll past the end of the program listings, a dummy entry, similar to the one shown in Figure 20-15, appears. The program count, explained in Subsection 3. below, increments to include this extra entry. The dummy entry, explained in Subsection 1. below, is simply a place holder in the program listing.

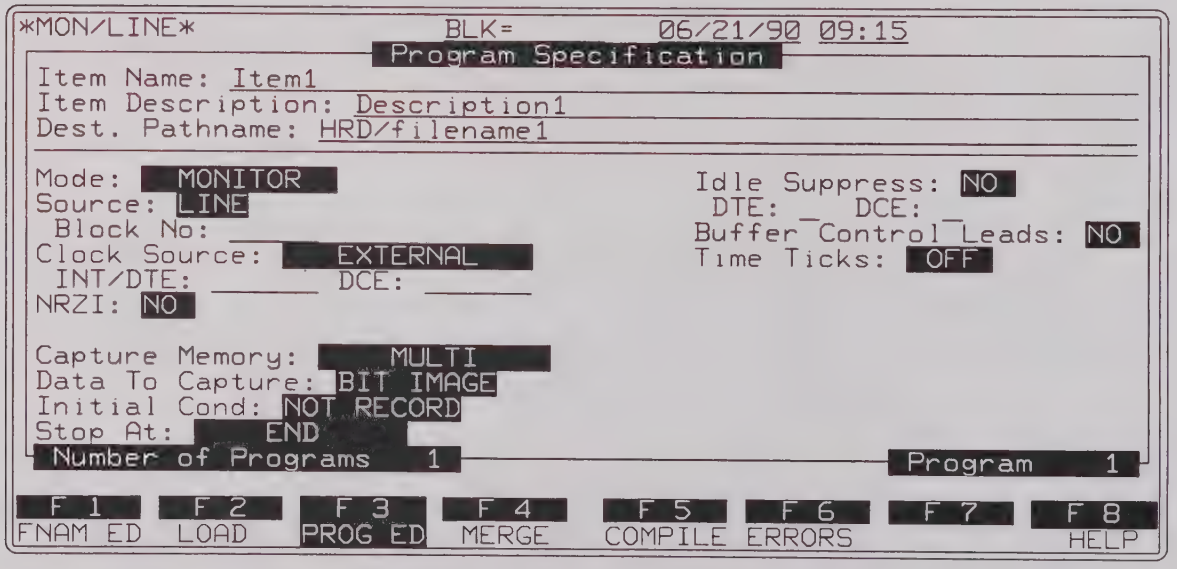


Figure 20-15 Default Program Specification Editor screen.

1. *Program identification fields*. The three fields in the top portion of the screen that identify a program—**Item Name**, **Item Description**, and **Dest. Pathname**—are text–entry fields.

These fields use the same editing keys as the Filename Specification screen. (See Section 20.4(C), above.) There are a few additional keys, however, that have specific functions on this screen:

SAVE	Saves the screen information (and also the location in the Easy View menus) about an entry in memory so that it can be recalled via the LOAD key.
CTRL – INSERT LINE	Inserts a dummy entry ahead of the entry containing the cursor. Use the dummy entry to hold a place for your new version in the program listing. Do not directly edit dummy entries.
LOAD	Restores a new numbered version of the entry last saved via the SAVE key. (The entry currently displayed on the screen will be overwritten.) Repeat to load as many versions as you need. Edit these extra copies to customize your own versions.
CTRL – DELETE LINE	Deletes the entry containing the cursor from the program listing. Deleting an entry from this listing does *not* delete it as a menu selection from the Easy View menus.

(a) *Item name.* Use the **Item Name** field to locate the program you want to modify. The item name is the menu selection as it appears in the Easy View menus.

> **NOTE:** If you loaded *mstrmenu.txt*, there are numerous entries. To delete an entry, press CTRL – DELETE LINE. Deleting an entry from the Program Specification screen, however, does *not* delete it from the Easy View menus.

There are two methods for creating a new version of one of the program listings. In the first method, simply overwrite the fields of a program entry (not a dummy entry) shown in the display.

The second method keeps the original version in the listing and adds a new version. First press SAVE to keep the original entry in memory. Then press CTRL – INSERT LINE to insert a place holder in the program listings and in the menus for your new version. Now press LOAD to overwrite the dummy entry with a numbered version of the original. Make any modifications to this entry. (The original is still in the listings. Press NEXT PAGE to view it.)

For either method, type in the new item name as you want it to appear on the Easy View menu. You will not be able to enter a name that exceeds the maximum length of 60 characters.

The new menu item is present when you return to the Easy View menus. It is located at the end of the same menu as the original entry.

(b) *Item description.* Use the **Item Description** field to enter some descriptive text about the menu entry. This text will be displayed toward the bottom of the Easy View menu screen whenever the selection bar is over the menu item. You will not be able to enter more than 62 characters.

(c) *Dest. pathname.* The pathname of the file associated with the **Item Name** is displayed in this field. It is the program that is actually loaded and run when you select the **Item Name** on an Easy View menu.

Type in any valid INTERVIEW pathname, including a drive specifier—*HRD*, *FD1*, or *FD2*. If you omit the drive specifier, an error message is displayed. You will not be able to enter a name that exceeds the maximum length of 126 characters (including the drive specifier). For assistance in file naming conventions, see Section 20.7 (also Section 15).

2. *Setups.* The fields in the bottom portion of the Program Specification screen display the current setup parameters for the file specified in the **Dest. Pathname** field. Modify these fields for your new version.

View the available selections in a setup field by pressing ROLL FWD and ROLL BACK. Move to the next field by pressing RETURN, ↓, or →. Use ↑ or ← to return to a previous field.

Most of these menu fields function as explained elsewhere in this manual:

For information on:	Refer to Section:
Line Setup	5
Record Setup	7
FEB Setup	9

A few of the setup fields on the Program Specification screen are a combination of two or more fields from the related Program–mode setup screen:

- **Source.** This field combines the **Source:** DISK and **Disk No.** fields of the Line Setup menu. Select from LINE, FD1, FD2, and HRD.

- **Capture Memory.** This field combines the **Capture Memory:** DISK and **Disk No.** fields and the **Capture Memory:** RAM and **Record Speed** fields of the Record Setup menu. Select from FD1, FD2, HRD, MULTI, NORMAL RAM, and HIGH SPEED RAM.

- **Time Ticks.** This field combines the **Time Ticks:** ON and **Tick Rate** fields of the Front–End Buffer Setup menu. Select from OFF, 1SEC, 100mS, 10mS, 1mS, 100uS, and 10uS.

All fields in the Setup portion of the screen are always present. There are, however, a few fields specific to selections made in the field that immediately precedes them:

- **Block No:** Enter a number only when **Source** is FD1, FD2, or HRD.
- **INT/DTE:** and **DCE:** Enter a speed only when **Clock Source** is INTERNAL or INTERNAL SPLIT. For the **INT/DTE:** field, you enter *either* an **INT**ernal speed (when the **Clock Source** is INTERNAL) *or* a **DTE** speed (when the **Clock Source** is INTERNAL SPLIT). The **DCE:** field does not apply when **Clock Source** is INTERNAL.
- **DTE:** and **DCE:** Enter an idle character only when **Idle Suppress** is YES.

3. *Program count.* Notice the two highlighted boxes in the lower corners of the screen. The box on the left tells how many programs are referenced in the **Source** file. This count increments (or decrements) when you insert (or delete) entries from the listing.

   The box on the right indicates the position within the program listing of the current entry. As you scroll through the program listings, the program number in the right–hand box changes.

(F) Merge Command

Use this command to merge the **Source** script file (and any new versions of its program setups made via the Program Specification screen) into the **Destination** file (named on the Filename Specification screen). If you are incorporating information into the Easy View menus, *HRD/MENU/mstrmenu.txt* is the **Destination** script file.

> **NOTE:** If you are merging your own script files, make sure the script follows the format explained in Section 20.5.

The MERGE operation uses dates within the **Source** and **Destination** script files to determine whether or not items in the destination file should be updated. See Section 20.5. If dates in the new (source) script file are more recent than dates in the old (destination) script file, merging is performed.

If, however, the new dates are the same as or earlier than the old ones, items in the destination file remain intact.

> **NOTE:** The date of the loaded script file is assigned to any modifications you make on the Program Specification screen.

A rotating arrow in the lower, right–hand corner of the display indicates that the merge is in progress. Press BRK if you want to abort the MERGE operation.

(G) Compile Command

This command takes the specified menu file, compiles it, and saves it to the pathname indicated on the Filename Specification screen. By default, the compiled file is *HRD/MENU/mstrmenu.cmp*.

A rotating arrow in the lower, right–hand corner of the display indicates that the compilation is in progress. Press [BRK] if you want to abort the COMPILE operation.

> *Caution: Only use the* COMPILE *command in the Easy View Maintenance utility program to compile* mstrmenu.txt. *Do not use the (C–compiler) File Maintenance Compile command.*

(H) Errors Screen

If errors occur during the execution of the LOAD, MERGE, or COMPILE command, you are instructed to go to this screen. To scroll through the list of errors, press [FREEZE]. Then use the cursor keys, [PREV PAGE], [NEXT PAGE], [↑], and [↓], to move through the buffer. [SHIFT]–[↑] moves you to the beginning of the buffer, [SHIFT]–[↓] to the end. Press [FREEZE] again when you leave this display.

The beginning and end of the current error list are delimited by the time the errors were written to the buffer. Additional error listings are appended to the end of the buffer. Once the buffer is full, the oldest error messages are forced out to accommodate the new ones.

(I) Help

Press the HELP softkey from any Easy View Maintenance screen to return to the main Help menu.

20.5 Script File Format

The Easy View utility FD1 Install File lets you view the contents of *FD1/MENU/install.txt*, the script file for AR–supplied application programs. Position the selection bar over FD1 Install File on the Easy View Utilities, Model and Demonstration Programs menu and press [RETURN]. All script files, including *mstrmenu.txt*, follow the same format you see for *install.txt*.

You may create your own script file on the INTERVIEW spreadsheet. Return to Program mode via the [EZ VU] key. From the main Program menu, press SPDSHT, then [EDIT]. Once you have completed your script file following the format described in this section, use the spreadsheet editor write command to save the file to disk: first, mark the text as a block; then press BLOCK, IN/OUT, WRITE/U, and then enter the pathname of the file. Press [XEQ]. For more information on the Spreadsheet editor, refer to Section 30.

There are three categories of information within script files: master information, menu information, and menu–item information. See Figure 20-16 for the general format of a script file and Section 20.5(D) for sample script files.

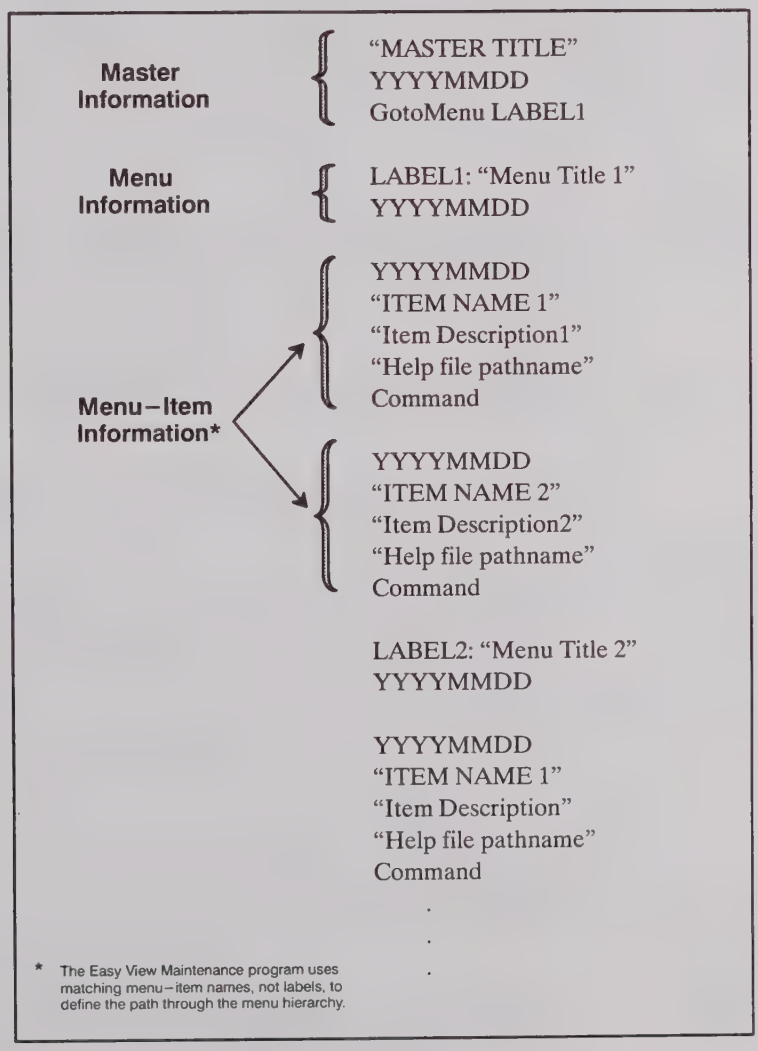


Figure 20-16 Script–file format.

(A) Master Information

Master information relates to the Easy View system as a whole. It includes a master title, date, and a command for the first menu to be displayed. Master information must be included at the top of every script file.

1. *Master title.* The top line of each Easy View menu screen contains a master title. By default, this line is blank. You can change this line by creating a new script file that contains a master title. (See the example in Section 20.5(D)1.)

   The master title has a maximum length of 64 characters. Enclose the title inside double quotes. The master title may include colons, but may not end with a colon. Otherwise, it will be interpreted as a label. See Section 3. below.

   The new title will automatically be centered at the top of Easy View menu screens.

   If you do not plan to change the master title, you can enter the title as empty quotation marks ("") in the source file. The master title in the destination file will not be updated, regardless of the master–title dates in the source and destination files.

   If the master title appears in the *destination* file as empty double quotes, however, the source master title will replace the destination title regardless of the master–title dates.

2. *Master–title date.* On the next line, enter the master–title date. During the merge operation, this source date is compared with the master–title date in the destination file.

   If the source date is more recent than the destination date, the source master title and master–title date will replace the destination title and date. If the source date is the same or earlier than the destination date, the destination title and date remain intact.

   The format for all dates in script files is YYYYMMDD (Y=year, M=month, D=day). Leading zeroes are ignored.

   You may enter the date as zero in the source file if you do not want to change the master title and date in the destination file. A zero in the destination file typically means that the title will be updated.

   > **NOTE:** Be careful when you assign dates in script files. Since the current date is usually sufficient to guarantee that items in a new file will update items in an older one, it's best not to use future dates. Unless carefully tracked, future dates may inadvertently prevent items from being updated.
   >
   > Also, note that dates are specific to the day only. They do not include hours, minutes, or seconds. This means that if you want to make an additional change to the same script–file item on the same day, you must advance the date by one day.

3. *GotoMenu LabelReference.* This is always the first command in the script file. It directs the system to display the menu identified by the label reference. If you are merging a script file with *mstrmenu.txt*, the information about the menu indicated here is *always* compared with the Easy View Main menu.

   The script file uses labels to connect menus together. In *mstrmenu.txt*, these labels appear as the word "MENU," followed by a string of digits and terminated by a colon (:). Notice, however, that in the GotoMenu command, the label *reference* does not end with a colon. Unique labels are *assigned dynamically each time a script file is merged with mstrmenu.txt*. You cannot assume that the labels are stable from one time to another.

   If you are creating your own label, begin it with a nonnumber and end it with a colon. For the corresponding label reference, simply omit the colon. During the merge operation, your label is automatically replaced by a new unique label that follows the format described above.

   > **NOTE:** If you are directly editing *mstrmenu.txt* (see Section 20.5(E)), use the current numbering scheme when referencing menu labels. (A printout of the file may help you track labels.)

(B) Menu Information

All Easy View menus are identified by labels in *mstrmenu.txt*. These labels are used to connect menus together. Matching *menu items*, not labels and label references, however, define the path through the menu hierarchy.

Each Easy View menu is entered from a *single* menu selection. If you want more than one menu selection to access the same menu, you must make duplicate copies of the menu, each with a unique label.

All script files must have at least one block of menu information. Menu information includes a menu title and a menu date.

1. *Menu title.* Each menu has an individual title with a maximum length of 58 characters. Enclose the title inside double quotes. The menu title may include colons, but may not end with a colon. Otherwise, it will be interpreted as a label. See Section 20.5(A)3., above.

   Menu titles are automatically centered on the menu–title line of Easy View menu screens.

   If you do not plan to change the name of an existing menu, you can enter the title as empty quotation marks (""). Make sure that you separate the label and the title, or empty quotes, by at least one space. The menu title in the destination file will not be updated, regardless of the menu dates in the source and destination files.

2. *Menu date*. The menu date follows the menu title. Once a menu item (see Section (C), below) is matched during the merge operation, this source date is compared with the menu date in the destination file.

   If the source date is more recent than the destination date, the source menu title and menu date will replace the destination title and date. If the source date is the same or earlier than the destination date, the destination title and date remain intact.

   Use the date format and guidelines explained in Section 20.5(A)2.

(C) Menu–Item Information

Information about each selection on a menu follows the menu title and date. There is no limit to the number of selections that an Easy View menu may have (other than available memory or disk space). Since each Easy View menu is entered from a *single* menu selection, you may not have duplicate selections on the same menu.

Menu–item information includes an item date, item title, item description, the pathname for an associated help file, and a command for the action to be executed when the menu item is selected.

1. *Item date*. For each menu item, there is a menu–item date. Once an item name is matched during the merge operation, (see Subsection 2. below) this source date is compared with the item date in the destination file.

   NOTE: The only time this comparison does not occur is when the destination menu has an empty set of double quotes as a menu item. In this case, any unmatched source item will be considered a match for the quotes, regardless of the dates.

   If the source date is more recent than the destination date, the source menu item and menu–item date will replace the destination item and date. If the source date is the same or earlier than the destination date, the destination item and date remain intact.

2. *Item name*. Each menu item has a maximum length of 60 characters. Enclose the title inside double quotes. (If you want an empty space in a menu, enter an empty set of double quotes.) The item name may include colons, but may not end with a colon. Otherwise, it will be interpreted as a label. See Section 20.5(A)3.

   During the merge operation, the INTERVIEW looks for matching menu–item names between the source and destination script files. Matching *menu items*, not labels and label references, define the path through the menu hierarchy.

   If no match is found, it is assumed that a new menu selection should be created.

A set of empty double quotes as an item name in the source file is a match only for empty quotes in the destination file. It simply means that a space should be reserved in the menu.

If a menu item in the source script file matches one in the destination file, the item dates are compared.

3. *Item description*. Each menu item can have an associated line of descriptive text, automatically centered in the bottom box on the Easy View screen. This text is displayed whenever the selection bar is over the menu item. The text may contain up to 62 characters, including colons, but it may not end with a colon. Otherwise, the text will be interpreted as a label. See Section 20.5(A)3.

   If you do not want to include descriptive text about a menu selection, enter an empty set of double quotes ("").

4. *Help–file pathname*. If you want to make a help file accessible when the selection bar is positioned over the menu item, enter the file's pathname. The pathname does not have to be enclosed in double quotes, but it may be. Filenames are case sensitive and have a maximum length of 126 characters (including the drive specifier). Absolute pathnames are preferred, although relative pathnames are permissible. Relative pathnames are assumed to be relative to the location from which the text menu file was loaded.

   If the pathname includes directories that do not currently exist in the filing system, the INTERVIEW automatically creates them.

   If you do not want to include a help file for a menu selection, enter an empty set of double quotes ("") for the pathname. If the associated item name is matched during the merge operation, any existing help–file pathname in the destination file remains intact.

   See Section 20.6 for information on creating a help file.

5. *Command*. The last piece of information required for each menu item is a command instructing the system what to do when RETURN is pressed to execute the menu selection. There are four possibilities.

   (a) *GotoMenu LabelReference*. The GotoMenu command directs the system to display the menu identified by the label reference.

   The script file uses labels to connect menus together. Matching *menu items*, not labels and label references, defines the path through the menu hierarchy.

   In *mstrmenu.txt*, labels appear as the word "MENU," followed by a string of digits, and terminated by a colon (:). Notice, however, that in the GotoMenu command, the label *reference* does not end with a colon. Unique

labels are *assigned dynamically each time a script file is merged with mstrmenu.txt*. You cannot assume that the labels are stable from one time to another.

If you are creating your own label, begin it with a nonnumber and end it with a colon. For the corresponding label reference, simply omit the colon. During the merge operation, your label is automatically replaced by a new unique label that follows the format described above.

(b) *ViewFile pathname*. The ViewFile command directs the system to display the help or information file identified by the pathname. The pathname does not have to be enclosed in double quotes, but it may be. Filenames are case sensitive and have a maximum length of 126 characters (including the drive specifier). Although absolute pathnames are preferred, relative pathnames are permissible. Relative pathnames are assumed to be relative to the location from which the text menu file was loaded.

(c) *ExecuteProgram pathname*. The ExecuteProgram command directs the system to load, compile, and run the program identified by the pathname. The file may be a program (PRGM), linkable–program (LPGM), or object (OBJ) file. (The rules governing hardware and/or software compatibility of the program between units, however, still apply. Consult Section 15.3(F).)

The pathname does not have to be enclosed in double quotes, but it may be. Filenames are case sensitive and have a maximum length of 126 characters (including the drive specifier). Although absolute pathnames are preferred, relative pathnames are permissible. Relative pathnames are assumed to be relative to the location from which the text menu file was loaded.

Use the directory structure shown in Section 20.7 to guide you in choosing a directory for your program.

(d) *ExecuteUtility pathname*. The ExecuteUtility command—supported in Version 1.02 (or higher) of the Easy View Maintenance program—directs the system to load, compile, and run the program identified by the pathname. The file may be a program (PRGM), linkable–program (LPGM), or object (OBJ) file. (The rules governing hardware and/or software compatibility of the program between units, however, still apply. Consult Section 15.3(F).)

Utility programs are those which do not monitor some aspect of a data line. Instead, they perform some other function such as disk or file maintenance and typically are accessed from the Utilities, Model and Demonstration Programs menu. Easy View Maintenance and File Transfer Shell are two examples of utility programs. The line setup parameters for these programs cannot be modified via the on–line editing feature. (See Section 4.)

The pathname does not have to be enclosed in double quotes, but it may be. Filenames are case sensitive and have a maximum length of 126 characters (including the drive specifier). Although absolute pathnames are preferred, relative pathnames are permissible. Relative pathnames are assumed to be relative to the location from which the text menu file was loaded.

(D) Sample Script Files

Caution: Before you merge any script file with mstrmenu.txt*, make sure that you have a backup copy of* mstrmenu.txt.

1. *Change the master title.* Enter a script file that follows the format shown in Figure 20-17 if you just want to change the master title.

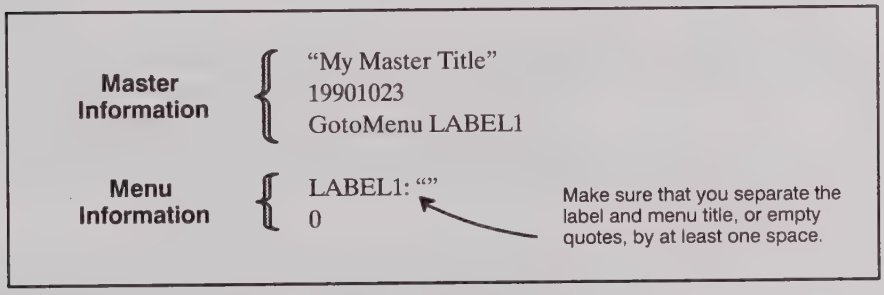


Figure 20-17 Minimum script file for adding a master title to the Easy View menus.

- Enclose your new title in double quotes on the first line of the script file.
- Enter the current date.
- Enter the GotoMenu command. The label reference can be anything, as long as it follows the syntax rules.
- At least one block of menu information must appear in every script file. Enter a label using the same name as referenced in the GotoMenu command above. Terminate the label with a colon. Normally, you would not change the title for an existing Easy View menu; leave a space after the label and then enter an empty set of double quotes and a zero for the menu–title date.

NOTE: If you use the format of this example, a blank entry is added to the end of the Easy View Main menu. If you do not want this extra entry, include an existing menu item in your script file. For example:

```
0
"AUTOMON"
""
""
ViewFile "FD1/usr/any_file"
```

Notice that the date is zero. This means that the existing menu–item information will not be updated. We suggest that you use the ViewFile command in the menu–item information since the GotoMenu command always causes a blank line to be added to some menu and the ExecuteProgram command requires the exact pathname of an existing program.

- Mark the text as a block and use the Spreadsheet editor WRITE/U command to save it as *FD1/MENU/install.txt*. See Section 30. Then access the Easy View Maintenance program and use the automatic installation feature to merge *FD1/MENU/install.txt* (source) with *HRD/MENU/mstrmenu.txt* (destination). See Section 20.3.

2. *Add a new selection to the Easy View Main menu*. Assume you have a program called *set_fkey* that assigns labels to the function keys in the Display Window. You want to add this program to the Easy View system. Figure 20-18 illustrates the script file you might create to install the program on the Easy View Main menu. (This example also incorporates a change to the master title.) The new entry will appear at the bottom of the Easy View Main menu.

   - Define the master title, if any, that will appear at the top of the Easy View menu screens.

   - Enter the GotoMenu command. The label reference can be anything, as long as it follows the syntax rules.

   - Enter a label for the Easy View Main menu using the same name as referenced in the GotoMenu command above. Terminate the label with a colon. The label, *LABEL1*, will be replaced by digits during the installation. Normally, you would not change the title for an existing Easy View menu; leave a space after the label and then enter an empty set of double quotes and a zero for the menu–title date.

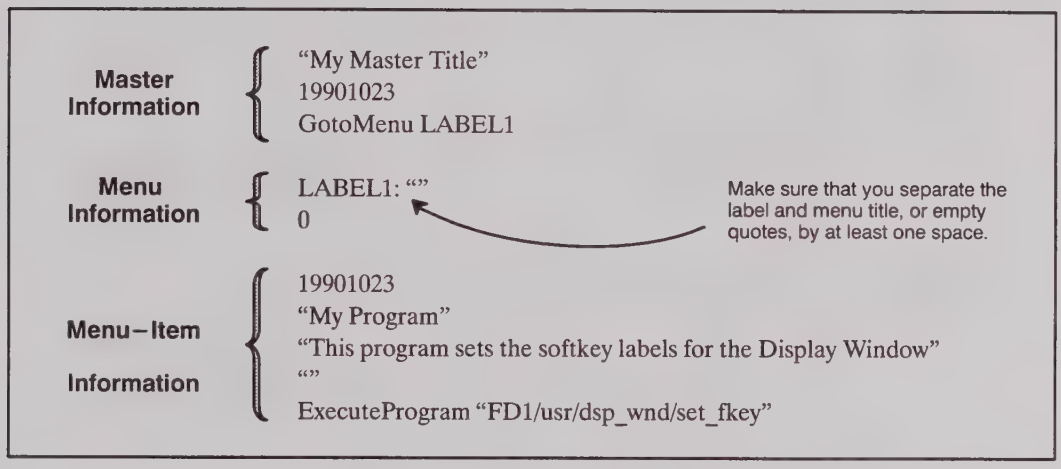


Figure 20-18 Script file for adding a master title and new selection to the end of the Easy View Main menu.

- Next, enter information about your new menu item. Use the current date. Then, enter the selection as you want it to appear on the Easy View Main menu: My Program.
- When the selection bar is over the new menu item, the descriptive text—*This program sets the softkey labels for the Display Window*—will appear in the bottom box of the menu screen.
- If you do not have a help file associated with the menu item, the help–file pathname is an empty set of quotation marks.
- Finally, specify the action to be taken when the selection bar is over My Program and you press RETURN: load, compile, and run the program in *FD1/usr/dsp_wnd/set_fkey*.
- Save the script file as *FD1/MENU/install.txt*. Then access the Easy View Maintenance program and use the automatic installation feature to merge *FD1/MENU/install.txt* (source) with *HRD/MENU/mstrmenu.txt* (destination). See Section 20.3.

3. *Add a new selection to the Menu of SDLC Emulation Programs.* Suppose you have an SDLC emulation program to add to the Easy View system. The Menu of SDLC Emulation Programs is a third–level Easy View menu.

   The script file begins with the master information (see Figure 20-19):

   - If you do not want to change the master title, enter empty quotation marks for the title (and a zero for the date).
   - Enter the GotoMenu command. The label reference can be anything, as long as it follows the syntax rules.

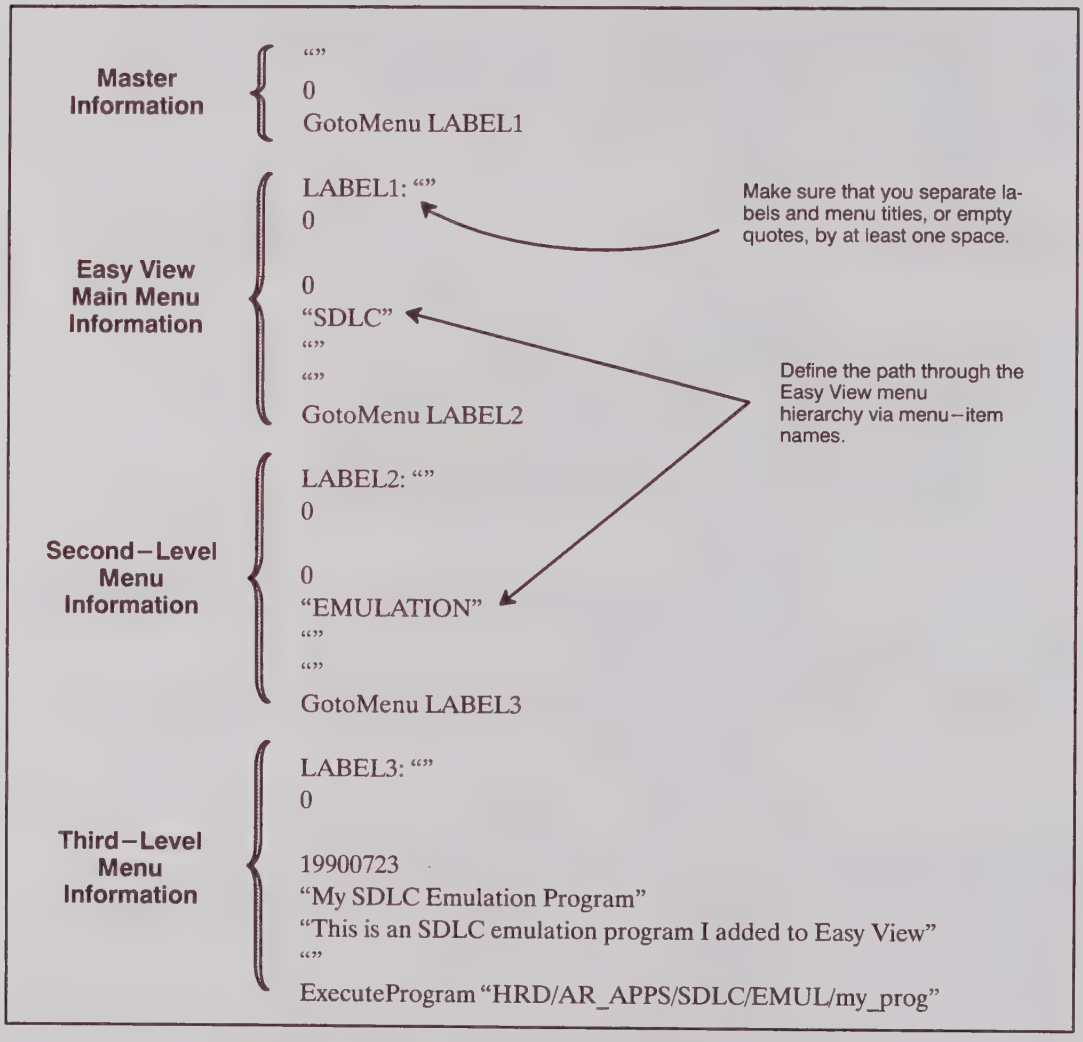


Figure 20-19 Script file for adding a new selection to a third–level Easy View menu.

Enter Easy View Main menu information:

- Enter a label for the Easy View Main menu using the same name as referenced in the GotoMenu command above. Terminate the label with a colon. Normally, you would not change the title for an existing Easy View menu; leave a space after the label and then enter an empty set of double quotes and a zero for the menu–title date.
- The path for accessing the Menu of SDLC Emulation Programs begins at the Easy View Main menu with the SDLC menu item. The Easy View Maintenance program uses matching menu–item names to move from menu to menu. So, enter "SDLC" as the menu item.

- There are no changes to this menu item, so enter empty quotation marks for the descriptive text and help file pathname, and enter a zero for the menu–item date.
- Enter the GotoMenu command. Enter a new label reference for the second–level Menu of SDLC Program Types, Tutorial(s) & How To's.

Enter second–level menu information:

- Enter a label for the Menu of SDLC Program Types, Tutorial(s) & How To's using the same name as referenced in the GotoMenu command above. Terminate the label with a colon. Normally, you would not change the title for an existing Easy View menu; leave a space after the label and then enter an empty set of double quotes and a zero for the menu–title date.
- The path for accessing the Menu of SDLC Emulation Programs continues from the Menu of SDLC Program Types, Tutorial(s) & How To's with the EMULATION menu item. So, enter "EMULATION" as the menu item.
- There are no changes to this menu item, so enter empty quotation marks for the descriptive text and help file pathname and a zero for the menu–item date.
- Enter the GotoMenu command. Enter a new label reference for the third–level Menu of SDLC Emulation Programs.

Enter third–level menu information:

- Enter a label for the Menu of SDLC Emulation Programs using the same name as referenced in the GotoMenu command above. Terminate the label with a colon. Normally, you would not change the title for an existing Easy View menu; leave a space after the label and then enter an empty set of double quotes and a zero for the menu–title date.
- Next, enter information about your new menu item. Use the current date. Then, enter the selection as you want it to appear on the Easy View Main menu: My SDLC Emulation Program.
- When the selection bar is over the new menu item, the descriptive text—*This is an SDLC emulation program I added to Easy View*—will appear in the bottom box of the menu screen.
- If you do not have a help file associated with the menu item, the help–file pathname is an empty set of quotation marks.
- Finally, specify the action to be taken when the selection bar is over My SDLC Emulation Program and you press RETURN: load, compile, and run the program in *HRD/AR_APPS/SDLC/EMUL/my_prog*. (Notice that the program is located in the Easy View directory structure. See Section 20.7.)

- Save the script file as *FD1/MENU/install.txt*. Then access the Easy View Maintenance program and use the automatic installation feature to merge *FD1/MENU/install.txt* (source) with *HRD/MENU/mstrmenu.txt* (destination). See Section 20.3.

(E) Editing *mstrmenu.txt*

Although *not* recommended, it is possible to make additional changes to the Easy View menus by directly editing *mstrmenu.txt*. If you do this, however, *make sure that you have a backup copy before you begin*.

> **NOTE:** Be aware that during an automatic installation of optional programs or updates to Easy View (from AR), it is assumed that the order of the *original* menu hierarchy still exists. If you have customized the menus, the installation process may restore necessary Easy View menus for the new program(s). If you want the new program to be installed in your customized menus, you must edit the install file that resides on the optional program disk.

1. *File Transfer Shell.* The preferred method for editing *mstrmenu.txt* is to transfer it out of the INTERVIEW as text via the File Transfer Shell program. This program is selected via the Easy View Utilities, Model and Demonstration Programs menu.

   Select **Text File:** YES and make the transfer via null modem to a local PC running a terminal–emulation program that supports XModem, YModem, and YModem Batch transfer.

   After the transfer is complete, exit the terminal–emulation program on the PC and save the text to a file with any name you wish. Use the PC's word processing to make your desired changes, and then save them to an ASCII text file. (Keep a backup copy of your revised file on the PC.)

   > **NOTE:** Although the labels will be changed when you compile *mstrmenu.txt*, use the current numbering scheme to reference menu labels during editing. (A printout of the file may help you track labels.)

   Return to the terminal–emulation program and transfer the text back to the INTERVIEW. You must use YModem Single or XModem for the transfer in order for the INTERVIEW to import the file as (lowercase) *mstrmenu.txt*.

   Run the Easy View Maintenance program. Press the COMPILE softkey (or the spacebar and C) to compile this new version of *mstrmenu.txt*. It is automatically saved as *mstrmenu.cmp*.

2. *INTERVIEW spreadsheet*. You can also edit *mstrmenu.txt* on the INTERVIEW's Protocol Spreadsheet. From Easy View, press [EZ VU], [PROGRAM], and the SPDSHT softkey to access the spreadsheet. To read *mstrmenu.txt* into the spreadsheet buffer, press [EDIT], BLOCK, IN/OUT, READ/U. Type in the name of the file, *HRD/MENU/mstrmenu.txt*.

   Although the labels will be changed when you compile *mstrmenu.txt*, use the current numbering scheme to reference menu labels during editing. (A printout of the file may help you track labels.)

   Once you've finished editing *mstrmenu.txt*, mark the entire text as a block. (The block is marked in reverse–video when you read the file into the spreadsheet. If you did not clear the highlighting, the block is still marked.) Then write it back to disk by pressing BLOCK, IN/OUT, WRITE/U. Type in the name of the file, *HRD/MENU/mstrmenu.txt*.

   Return to Easy View via the [EZ VU] key. Access the Easy View Maintenance program. When the logo screen appears, press the spacebar. The only function you need to perform is the compilation of *mstrmenu.txt*, so press [C] (or the COMPILE softkey) to compile this new version of *mstrmenu.txt*. It is automatically saved as *mstrmenu.cmp*.

20.6 Creating a Text (Help) File

The Easy View help and information files are normally INTERVIEW files of type ASCII. Since Easy View does not check the file type, however, any files which contain ASCII text may be used.

To make your own help files accessible from the Easy View system, reference them in the menu–item information of your script file. See Section 20.5(C).

(A) Format

Easy View text files must use the following format.

- Files are limited to 32,767 characters. If a longer file is encountered, only the first 32,767 characters in it are used.
- File should contain only printable ASCII characters and new line (line feed) characters.
- Format the text as 64–character lines terminated by a new line character. The 64 characters *excludes* the new line. Lines exceeding 64 characters are truncated when they are displayed on the screen.
- The page length on the INTERVIEW screen is 20 lines. If you restrict your pages to this length, you can use the [NEXT PAGE] and [PREV PAGE] keys to easily scroll through the file.
- Text file pathnames have a maximum length of 126 characters *including* the disk drive specifier. Choose from the following drive specifiers: *HRD*, *FD1*, and *FD2*.

When the menu system tries to open a text file, it uses the drive specifier in the file's pathname to determine where it should search for the file. The following table shows which drives are searched and the order in which they are accessed. The same strategy is followed when running a program from a menu.

Pathname Starts With	Order of Search
HRD	*HRD*
FD1	*FD1, FD2*
FD2	*FD2, FD1*

20.7 Easy View Directory Structure

The directory structure for the Easy View system is similar to the menu structure. See Figure 20-20.

All of the AR–supplied application programs that you can load and run from an Easy View menu (and their associated help files) are located in one of the subdirectories of the */AR_APPS* directory on the hard disk. The subdirectories are named by protocol and are similar to the Easy View Main menu selections. Find the appropriate protocol subdirectory and change directories. The next set of subdirectories groups the programs according to the function they perform: emulation, monitoring, statistics, or conformance.

The script file for the Easy View system and the compiled menu–information file reside in the */MENU* directory on the hard disk.

How To files and tutorials reside in the */TUTORIAL* directory of the hard disk.

The */sys* and */usr* directories may include programs, *#include* files, and setups. These two directories also contain the INTERVIEW system software.

You may want to follow these file–naming conventions:

- Program names use upper and lower case as needed to improve readability. The file maintenance system is case sensitive. When referencing files, but sure to enter the names exactly as they appear in the directories.

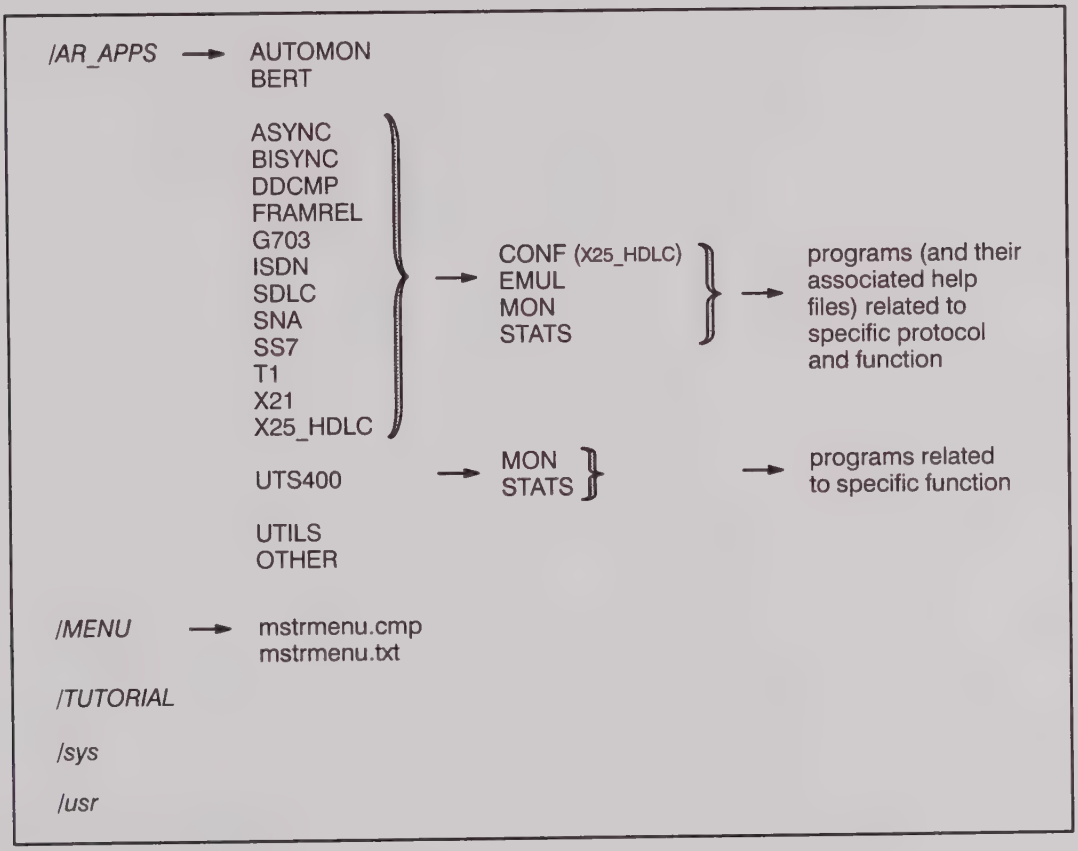


Figure 20-20 Easy View directory structure on the hard disk.

- Program names are limited to ten characters, if you want to use one of the extensions listed below. Otherwise, they may be up to 12 characters in length.
- Object (OBJ), linkable–program (LPGM), and other run–time source files do not have extensions. Run–time source files are those which contain everything needed for the program to run in a single file.
- Otherwise, an extension identifies the contents of a file.

.d	document/help for applications (*.d* files formatted for a 20–line display reside in the same directory as their associated program files)
.t	tutorial on protocols or other non–INTERVIEW related subject
.i	instructions/help related to operation of the INTERVIEW
.s	spreadsheet source (not run–time source)
.c	C–language source
.h	C–language header (*#include* file)
.o	linkable–object file

21 Tabular Statistics

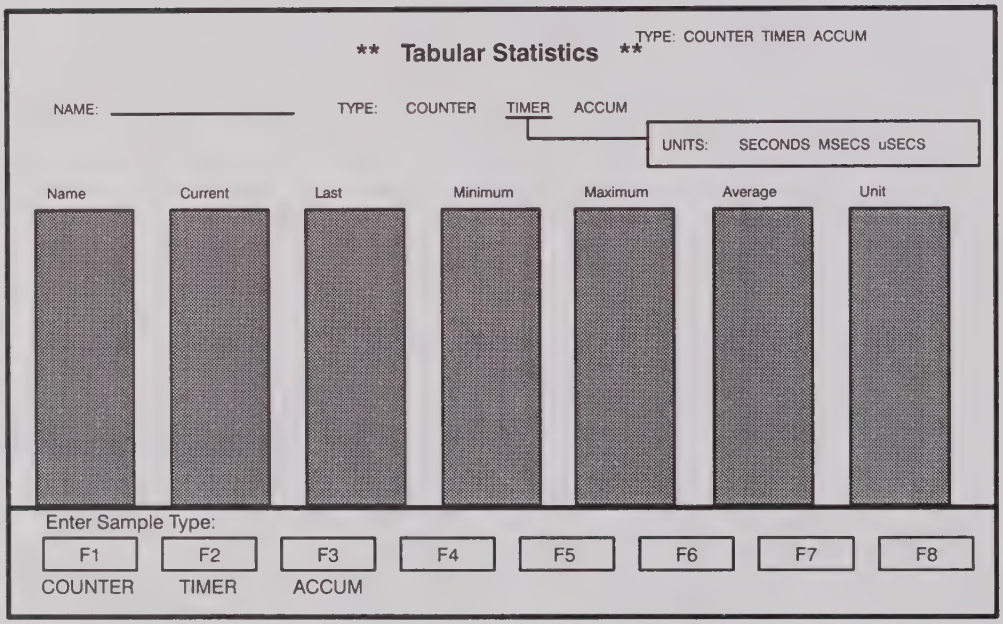


Figure 21-1 Menu fields, Tabular Statistics screen.

21 Tabular Statistics

The users of the INTERVIEW can assign tasks easily to an almost unlimited number of software counters and timers. When these incrementing counters and timers are sampled—that is, when they are read and cleared—their current totals are factored into a statistical breakout on the Tabular Statistics screen. This breakout is a real−time reading of current, last, minimum, maximum, and average values for the counter or timer.

At any one moment, the Tabular Statistics screen displays a maximum of 75 values for 15 counters, timers, and accumulators. (Accumulators are defined below in Section 21.4). The Graphic Statistics screen, treated in the next section, displays less information (16 values total) at any one time, but in a graphic format. Both statistics screens can be scrolled up or down to display additional rows of values.

An additional statistics screen can also be accessed via the STATS softkey. Press BERT to view the Run−mode BERT statistics display. This screen is discussed in conjunction with the BERT Setup procedures; refer to Section 11.9.

The role of triggers in creating, operating, sampling, and accumulating various counters and timers is common to both screens and will be discussed here under Tabular Statistics.

21.1 Counters and Timers

Counters and timers are operated as *actions* on trigger menus and in Protocol Spreadsheet tests. In the example below, two different counters are made to increment as spreadsheet actions.

In this Bisync example, polling address A represents a drop on a multipoint circuit. The string "^x A" on the DCE side is the beginning of a text block originating at remote drop A. When the spreadsheet program sees this string, it moves to a state called drop_a, where the end of every text block (DCE STRING "^F") increments one counter (allblk_a) and only blocks that end with a bad BCC increment another counter (badbcc_a).

```
LAYER: 1
    STATE: stx
        CONDITIONS: DCE STRING "^x A"
        NEXT_ST: drop_a
    STATE: drop_a
        CONDITIONS: DCE BAD_BCC
        ACTIONS: COUNTER badbcc_a INC
```

```
CONDITIONS: DCE STRING "F_F"
ACTIONS: COUNTER allblk_a INC
NEXT_ST: stx
```

The current value of a counter also can be used as a *condition* either on a trigger menu or in a Spreadsheet test. Here is an example of a counter performing this "countdown" function:

```
CONDITIONS: COUNTER allblk_a EQ 1000
ACTIONS: COUNTER badbcc_a SAMPLE
```

(The SAMPLE action is explained in Section 21.3, below.)

Timers are not used as trigger conditions, since timeouts serve this function.

21.2 Preparing the Tabular Statistics Screen

Current values of counters and timers are read on the Tabular Statistics and Graphic Statistics screen. Both statistics screens are always accessible by softkey during Run mode. A counter or timer that is named in a trigger must be identified by name on the statistics screen. This naming is done in Program mode prior to the run.

Press [PROGRAM] and then [F4] for the Statistics Menu screen. Press [F1] (or [RETURN]) to enter the Tabular Statistics screen. In Program mode, the screen shows 15 tabular rows beneath a single line of menu fields. There are two cursors, one on the menu line and one in the table. See Figure 21-2. The fields on the menu line (second line at the top of the screen) always refer to the row in the results table that has the lower cursor.

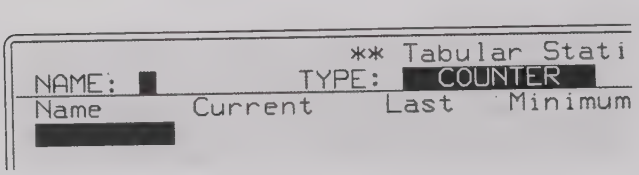


Figure 21-2 In Program mode, this screen has two cursors.

When you enter the Tabular Statistics screen, the upper cursor is in the **Name** field on the menu line, while the lower cursor is in Row One of the table.

Press [DONE] or [RETURN] (or [→] and [←]) to move the upper cursor from field to field in the menu area of the screen. Press [ROLL FWD] and [ROLL BACK] to change the selections in rotating–window fields.

Press [↓] and [↑] to move the lower cursor from display line to display line. The menu selections at the top of the screen will change as you cursor down the screen, since they are always keyed to the display line that has the cursor.

Each time the cursor advances one row down the table, the information that the user has entered on the menu line is written to the previous row in the table. In the top half of the sequence in Figure 21-3, the user has entered a name on the menu line above a blank table. The cursor is now in the **Type** field, where COUNTER was the default selection. The bottom of the figure shows the resulting table after the user presses ↓ .

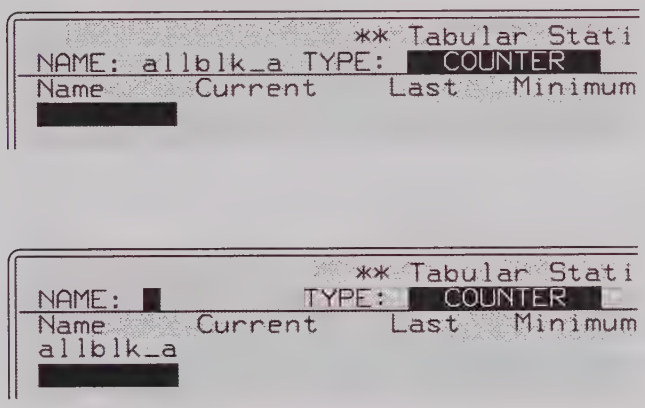


Figure 21-3 When lower cursor moves, user data is written to the vacated display line.

↑ reverses the direction of the lower cursor. The user may name or revise counters, timers, and accumulators by moving up the table as well as down.

The tabular area of the screen is a scrolling display of variable length that sets a very high limit (100) on the number of counters, timers, and accumulators that can be named by the user. To scroll down the directory, position the cursor on the last line of the listings and press ↓. This keystroke will display new lines one at a time. Or press NEXT PAGE to display 15 new lines of counters, timers, and accumulators. SHIFT and ↓ together move the cursor to the end of the listings.

Position the cursor at the top of the listings and press ↑ to expose lines that have scrolled off the screen at the top. Or press PREV PAGE to retrieve an entire previous page of listings. SHIFT – ↑ will always move you to the top of the listings.

INSERT LINE and DELETE LINE are operative keys on the scrolling statistics tables.

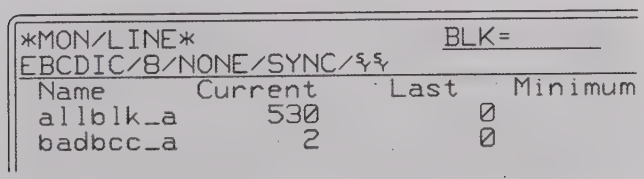


Figure 21-4 All counting and timing is performed in the Current column.

When RUN is pressed, a counter or timer that has been named in a trigger action will show its current value next to its name on the statistics screen. Figure 21-4 is a Run−mode display of two counters, one that is incrementing with each text block sent by a particular remote drop, and a second that is incrementing with each bad BCC from the same source.

If you have named a counter (or timer or accumulator) in a trigger action but forgotten to identify it on a statistics screen, the statistics will still be available in Program mode following the run (provided you have *sampled* the counter or timer at some point during the run). To view the statistics, simply identify the counter (timer, accumulator) by name on the statistics screen and move the lower cursor. The statistics from the previous run will appear on the screen next to the name.

21.3 Sampling Current Values

In addition to current value, the Tabular Statistics screen has columns for last, minimum, maximum, and average values. See Figure 21-5. Unit is not a value column. It applies to timers only, and reflects the unit of time—second, millisecond, or microsecond—selected by the user for that timer on the menu line during Program mode.

Last, **Minimum**, **Maximum**, and **Average** are statistical columns, based on previous samplings of the Current column. Sampling is a trigger action that reads the current value of the counter or timer and then resets it to zero. The **Last** column receives the sampled value. The other columns—**Minimum**, **Maximum**, and **Average**—compare the sampled value with previous samples.

```
*MON/LINE*                    BLK=        06/16/89 13:17
EBCDIC/8/NONE/SYNC/$$
 Name      Current     Last   Minimum Maximum  Average     Unit
 allblk_a      999        0
 badbcc_a        4        0
```

```
*MON/LINE*                    BLK=        06/16/89 13:18
EBCDIC/8/NONE/SYNC/$$
 Name      Current     Last   Minimum Maximum  Average     Unit
 allblk_a        0     1000
 badbcc_a        0        4         4       4        4.00
```

Figure 21-5 The current count ends when the counter is sampled.

We have already seen a counter that incremented with every bad BCC. A Spreadsheet trigger that *sampled* this incrementing counter every 1000 blocks would maintain a statistical record of errored blocks per thousand:

```
LAYER: 1
    STATE: stx
        CONDITIONS: DCE STRING "^S A"
        NEXT_ST: drop_a
        CONDITIONS: COUNTER allblk_a EQ 1000
        ACTIONS: COUNTER badbcc_a SAMPLE
            COUNTER allblk_a SET 0
    STATE: drop_a
        CONDITIONS: DCE BAD_BCC
        ACTIONS: COUNTER badbcc_a INC
        CONDITIONS: DCE STRING "^F"
        ACTIONS: COUNTER allblk_a INC
        NEXT_ST: stx
```

In Run mode, zero appears in the **Last** column prior to the first sampling of a counter or timer, and nothing appears in the **Minimum**, **Maximum**, and **Average** columns. See the top of Figure 21-5. The bottom of the same figure illustrates the effect of the first sampling. The counter named badbcc_a is cleared automatically but its sampled value is retained in the Last column. Since this is a first sampling, the sampled value is carried over unchanged to the Minimum, Maximum, and Average columns also. Note that the counter named allblk_a was not sampled, so it had to be reset manually (COUNTER allblk_a set 0).

The next example uses a *timer* in an X.25 environment. A pair of triggers start and sample a timer called t2. Each sample is a measurement of the timeout observed by an X.25 PAD before it responds with an RR to a DCE Info frame. (This timeout is called T2 in X.25.) INFO and RR are spreadsheet conditions in the protocol package for X.25 Layer 2 (see Section 37.)

```
CONDITIONS: DCE INFO GDBCC
ACTIONS: TIMER t2 RESTART
CONDITIONS: DTE RR
ACTIONS: TIMER t2 SAMPLE
```

Figure 21-6 shows a set of results that might be generated by these two triggers.

```
*MON/LINE*                    BLK=          06/16/89 13:19
ASCII/8/NONE/BOP
 Name      Current     Last    Minimum Maximum  Average     Unit
 t2              6      110         15     120   111.83 MSECS
```

Figure 21-6 This timer has been sampled several times.

21.4 Accumulators

Accumulators look like counters and timers on the statistics screens but they do not increment or reset counters, nor do they start or stop timers. Rather, they *accumulate selected samplings* of these counters and timers without interfering with the counting and timing functions. Thus, they enhance the performance of counters and timers by empowering them to work for several accumulators at the same time.

For example, we have already designed a pair of counters that counted bad BCCs per thousand blocks with respect to one drop on a multipoint circuit. We will enlarge the program with a pair of counters for each of two additional drops, drop B and drop C. Then we will add an accumulator to generate error–per–thousand statistics for *the three drops taken together*.

Accumulating is a trigger action found on spreadsheet softkeys but not on trigger menus. The ACCUMULATE actions in our spreadsheet program might look like this:

```
LAYER: 1
    STATE: stx
        CONDITIONS: DCE STRING "%A"
        NEXT_ST: drop_a
        CONDITIONS: COUNTER allblk_a EQ 1000
        ACTIONS: COUNTER badbcc_a SAMPLE
            COUNTER allblk_a SET 0
        CONDITIONS: DCE STRING "%B"
        NEXT_ST: drop_b
        CONDITIONS: COUNTER allblk_b EQ 1000
        ACTIONS: COUNTER badbcc_b SAMPLE
            COUNTER allblk_b SET 0
        CONDITIONS: DCE STRING "%C"
        NEXT_ST: drop_c
        CONDITIONS: COUNTER allblk_c EQ 1000
        ACTIONS: COUNTER badbcc_c SAMPLE
            COUNTER allblk_c SET 0
    STATE: drop_a
        CONDITIONS: DCE BAD_BCC
        ACTIONS: COUNTER badbcc_a INC
            ACCUMULATE alldrop COUNTER badbcc_a CURRENT
        CONDITIONS: DCE STRING "^F"
        ACTIONS: COUNTER allblk_a INC
        NEXT_ST: stx
    STATE: drop_b
        CONDITIONS: DCE BAD_BCC
        ACTIONS: COUNTER badbcc_b INC
            ACCUMULATE alldrop COUNTER badbcc_b CURRENT
        CONDITIONS: DCE STRING "^F"
        ACTIONS: COUNTER allblk_b INC
        NEXT_ST: stx
    STATE: drop_c
        CONDITIONS: DCE BAD_BCC
        ACTIONS: COUNTER badbcc_c INC
            ACCUMULATE alldrop COUNTER badbcc_c CURRENT
        CONDITIONS: DCE STRING "^F"
        ACTIONS: COUNTER allblk_c INC
        NEXT_ST: stx
```

The statistics table now can show results for six counters and one accumulator (Figure 21-7). The accumulator gives last, minimum, maximum, and average error–per–thousand counts based on all the drops on the circuit.

MON/LINE BLK= 06/16/89 13:19
ASCII/8/NONE/BOP

Name	Current	Last	Minimum	Maximum	Average	Unit
allblk_a	516	1000				
badbcc_a	0	1	0	5	2.03	
allblk_b	801	1000				
badbcc_b	2	2	0	8	4.22	
allblk_c	391	1000				
badbcc_c	11	26	16	37	24.62	
alldrop		26	0	37	10.30	

Figure 21-7 The accumulator at the bottom of the table is consolidating errors–per–thousand values from three separate drops.

Not only current values but also last, minimum, and maximum values can be accumulated and broken out statistically. Values in the Maximum column, for example, often are significant *limit* values: time limits, size limits, and so forth. An accumulator might be assigned to sample only this maximum value for several counters or timers running concurrently. The resultant tabular row would be a comparison of these maximum values.

21.5 Keeping a Statistical Log

The sampling action can be used to log statistics at regular time–intervals. In the example that follows, the program counts data packets on an X.25 link and sends a line of date– and time–stamped statistical values every hour on the hour to a serial printer attached to the INTERVIEW. DTE DATA and DCE DATA are packet–level conditions in the protocol package for X.25 Layer 3 (see Section 38). PRINT COUNTER (and PRINT TIMER) is a layer–independent action described in Section 31.4.

```
LAYER: 3
   TEST: paks_per_hr
      STATE: six_am
         CONDITIONS: TIME 0600
         ACTIONS: TIMEOUT sixtysec RESTART 60
         NEXT_ST: hourly
      STATE: hourly
         CONDITIONS: TIMEOUT sixtysec
         ACTIONS: TIMEOUT sixtysec RESTART 60
            COUNTER minutes INC
         CONDITIONS: COUNTER minutes EQ 60
         ACTIONS: COUNTER minutes SET 0
            COUNTER datapaks SAMPLE
            PRINT COUNTER datapaks
         CONDITIONS: DTE DATA
         ACTIONS: COUNTER datapaks INC
         CONDITIONS: DCE DATA
         ACTIONS: COUNTER datapaks INC
```

After several hours, the resulting printout might look like this:

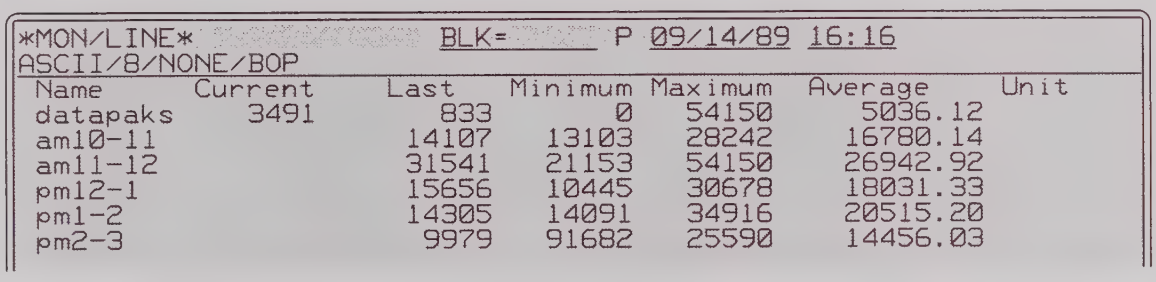
Time	Name	Current	Last	Minimum	Maximum	Average	Unit
09/14 07:00	datapaks	0	0				
09/14 08:00	datapaks	0	0	0	0	00.00	MSECS
09/14 09:00	datapaks	0	820	0	820	410.00	MSECS
09/14 10:00	datapaks	0	3388	0	3388	1402.67	MSECS

Figure 21-8 A counter is sampled every hour and its values are logged to a serial printer.

Accumulators might be added to the original program to gather statistics for a certain hour each day over a period of days or weeks:

```
TEST: time_of_day
    STATE: times
        CONDITIONS: TIME 1105
        ACTIONS: ACCUMULATE am10–11 COUNTER datapaks LAST
        CONDITIONS: TIME 1205
        ACTIONS: ACCUMULATE am11–12 COUNTER datapaks LAST
        CONDITIONS: TIME 1305
        ACTIONS: ACCUMULATE pm12–1 COUNTER datapaks LAST
        CONDITIONS: TIME 1405
        ACTIONS: ACCUMULATE pm1–2 COUNTER datapaks LAST
        CONDITIONS: TIME 1505
        ACTIONS: ACCUMULATE pm2–3 COUNTER datapaks LAST
```

The resulting tabular screen could tell you, for example, the average number of data packets traveling over the link between the hours of 11 A.M. and 12 noon for a given Monday through Friday:



MON/LINE BLK= P 09/14/89 16:16
ASCII/8/NONE/BOP

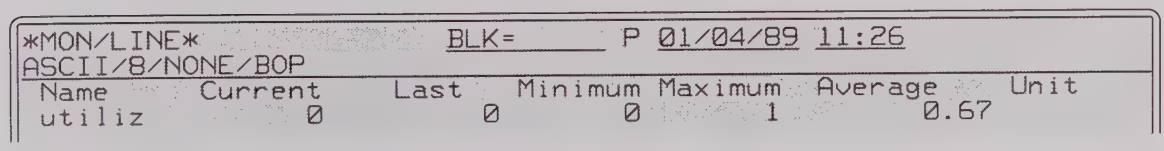
Name	Current	Last	Minimum	Maximum	Average	Unit
datapaks	3491	833	0	54150	5036.12	
am10-11		14107	13103	28242	16780.14	
am11-12		31541	21153	54150	26942.92	
pm12-1		15656	10445	30678	18031.33	
pm1-2		14305	14091	34916	20515.20	
pm2-3		9979	91682	25590	14456.03	

Figure 21-9 Here statistics are accumulated for specific hours of the day over a period of days or weeks.

21.6 The Sampling Action as Divisor

The sampling action can be used to divide the sum of all sampled current values on a counter by another value. To divide X by Y, count the events that add up to X and sample the counter Y times. The quotient or proportion will appear in the **Average** column of the counter.

Suppose, for example, that you want to divide the number of information frames on a link by *all* frames, as an indicator of how efficiently the link is being utilized. Count all Info frames on a counter called **info**. Sample the **info** counter whenever *any* frame is seen. Reference the **info** counter on the Tabular Statistics screen. If 1000 out of 1500 frames are Info frames, your results will look like those in Figure 21-10.



MON/LINE BLK= P 01/04/89 11:26
ASCII/8/NONE/BOP

Name	Current	Last	Minimum	Maximum	Average	Unit
utiliz	0	0	0	1	0.67	

Figure 21-10 The value in the *Average* column for this counter is the sum of all sampled values divided by the number of samples.

22 Graphic Statistics

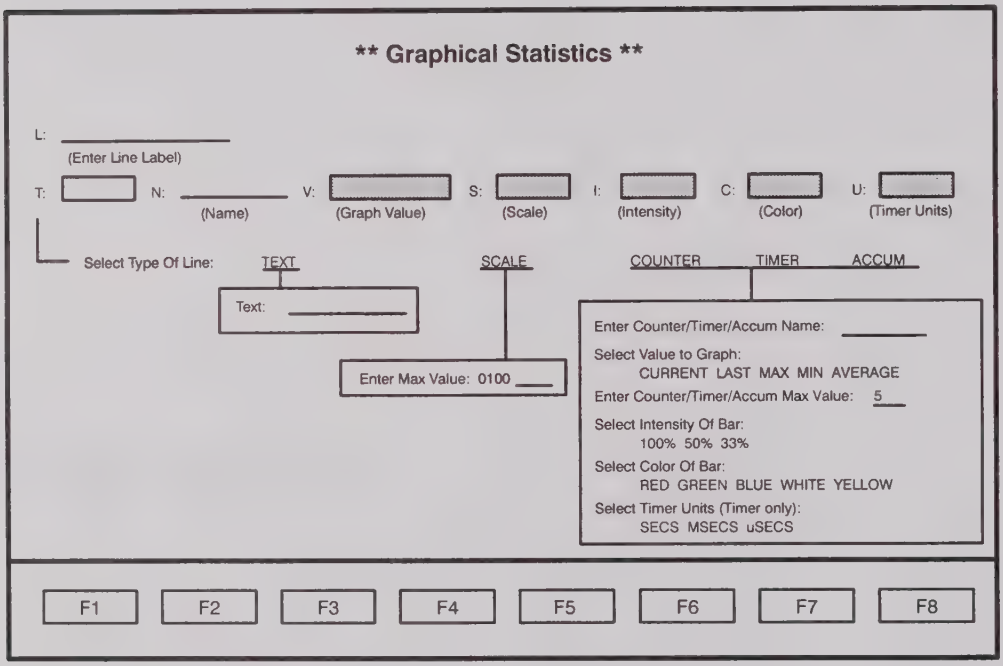


Figure 22-1 Setup selections on the Graphical Statistics screen.

22 Graphic Statistics

The operator of the INTERVIEW will find it easy to design a bar–graph display on the Graphical Statistics screen, with color parameters that can be mapped to a color monitor. Counters, timers, and accumulators that are referenced on the Graphical Statistics screen display their values on this screen as horizontal bars that are drawn in real time and retained in Freeze and Program modes. Variously shaded bars for up to 16 counters, timers, and accumulators may be displayed.

Any of the 16 horizontal lines in the graphics display may be reserved for explanatory text or scale numbers instead of a graphic bar. The bars themselves can represent statistics chosen from the entire pool of counter and timer values, grouped and renamed on the graphics display for clarity of overall presentation.

22.1 Enabling the Graphic Display

Both of the statistics screens, tabular and graphics, are enabled at all times during Run mode and can be entered via softkey. Both statistics displays are named on the second bank of softkeys in Run mode. See Figure 22-2. (On the first bank of Run mode softkeys, F5 is labeled STATS and will call up whichever screen is the current or dormant entry— TABULAR or GRAPHIC —in the **Statistics Type** field on the Display Setup menu.)

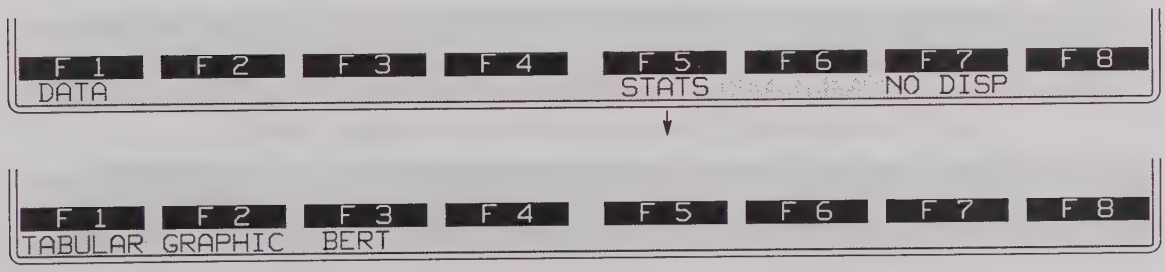


Figure 22-2 Both statistics displays can be entered via the second bank of softkeys during Run mode.

> **NOTE:** An additional statistics screen can also be accessed via the STATS softkey. Press BERT to view the Run–mode BERT statistics display. This screen is discussed in conjunction with the BERT Setup procedures; refer to Section 11.9.

No graphic bar is drawn until a counter or timer has been named on the Graphical Statistics screen in Program mode and then put in motion by the program during Run mode. Examples of triggers that control counters and timers are given in the preceding section, Tabular Statistics. After you have created your counters and timers in the trigger–menu or spreadsheet program, enter the Graphical Statistics screen by pressing PROGRAM, F4 for Stats, and F2 for Graphical Statistics.

22.2 Cursor Movement on Graphical Statistics Menu

There are always two cursors in the Graphical Statistics menu in Program mode. When you enter the screen, one cursor is in the L(abel) field in the menu area at the top of the screen and the lower cursor is on the top line of the 16−line display area. See Figure 22-3. The menu−field area always applies to the horizontal display line that has the lower cursor. In Figure 22-3, the display area is blank, and the L(abel) field and other menu fields are in default condition.

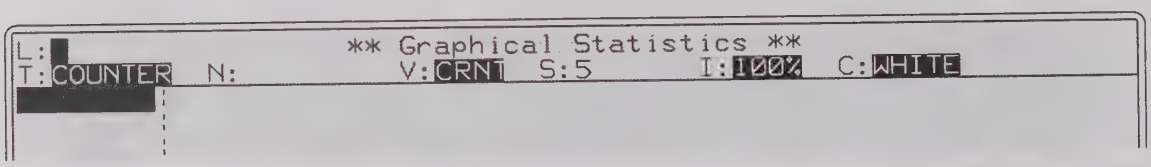


Figure 22-3 Two cursors on default graphics setup screen: the fields in the menu area always pertain to the display line that has the lower cursor.

Press [DONE] or [RETURN] (or → and ←) to move the upper cursor from field to field in the menu area of the screen. Press [ROLL FWD] and [ROLL BACK] to change the selections in rotating−window fields.

Press ↓ and ↑ to move the lower cursor from display line to display line. The menu selections at the top of the screen will change as you cursor down the screen, since they are always keyed to the display line that has the cursor. New menu−area selections and data entries are written to the display line that has the lower cursor as soon as that cursor is moved up or down.

The graphics area of the screen is a scrolling display of variable length that sets a very high limit (48) on the number of bar, scale, and text lines that may be created by the user. To scroll down the directory, position the cursor on the last line of the listings and press ↓. This keystroke will display new bar lines one at a time. Or press [NEXT PAGE] to display 16 new lines of bar lines. Pressing [SHIFT] and ↓ together, moves the cursor to the end of the listings.

Position the cursor at the top of the listings and press ↑ to expose lines that have scrolled off the screen at the top. Or press [PREV PAGE] to retrieve an entire previous page of listings. [SHIFT]−↑ will always move you to the top of the listings.

22.3 Menu Fields

(A) Label

L is the label field. The horizontal bar lines on the Graphical Statistics screen have labels at the far left. Referring to Section 22.3(B), below, give each bar a name that is compatible with the *text* line at the top of the chart and with the *scaling numbers* above or below the chart.

The label does not have to correspond to the *name* of the counter or timer (or accumulator) on the trigger menus or in the Protocol Spreadsheet program. For example, the label *PHILA* might be used for a counter named badbcc_a, if the counter is tracking errored blocks sent from multidropped device A in Philadelphia.

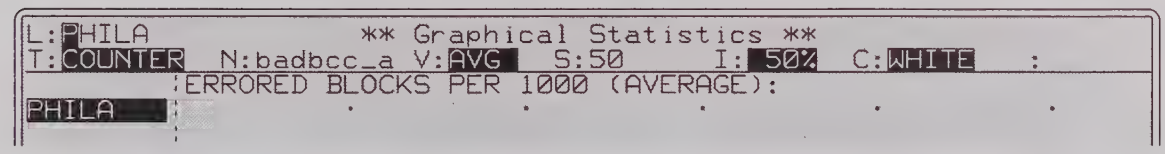


Figure 22-4 The label *PHILA* will appear alongside a horizontal bar representing the average value of the counter named "badbcc_a."

The **L** field is eight columns wide. Any ASCII entry may be made or a field may be left blank, as in the line of text *ERRORED BLOCKS PER 1000 (AVERAGE):* in Figure 22-4. A label in the **L** field is written to the display line that has the lower cursor as soon as that cursor is moved up or down.

(B) Type

T designates the type of horizontal line that will be created at the lower cursor. Line types are TEXT, SCALE, COUNTER, TIMER, and ACCUM. Counters, timers, and accumulators will be represented in Run mode by horizontal bars of various shadings that lengthen and shorten as the values for the counters, timers, and accumulators increase and decrease.

T: TEXT devotes a display line to explanatory text. The text line shown in Figure 22-4 was created by the **T:** TEXT entry in Figure 22-5. The text entry may be 54 characters long. This is the full width of the graphic display area.

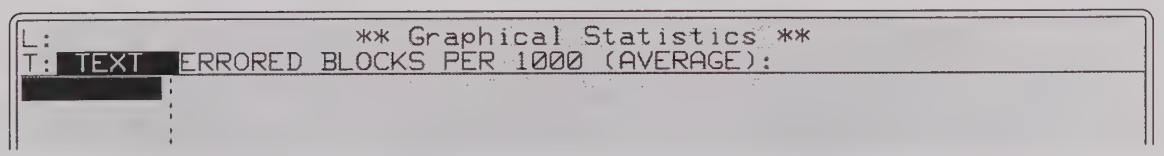


Figure 22-5 Explanatory text will be written to the display line that has the lower cursor as soon as that cursor is moved up or down.

T: SCALE creates a line with five scaling numbers. Enter a number in the **S** field that represents the highest number of units you will want to display on the graph. The entry may be placed anywhere in the **S** field. An example of an **S** entry and the scale line that results is given in Figure 22-6.

The logic will distribute the other four scaling numbers on the scaling line. It will scale the value you enter directly; or else it will raise your value to the next value that fits the scaling algorithm.

To be scaled directly, your **S** number should be expressible by a **single digit of precision** in scientific notation. The number 40, for example, will be applied directly to the scaling line, since in its scientific expression—4 x 10^1—4 is a single digit. 45 (4.5 x 10^1) will be raised to 50. Here is the beginning of the series of valid **S** numbers: 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, etc., up to and including 90,000. (If you enter a value between 90,001 and 99,999, the scaling logic will raise the value to 100,000.)

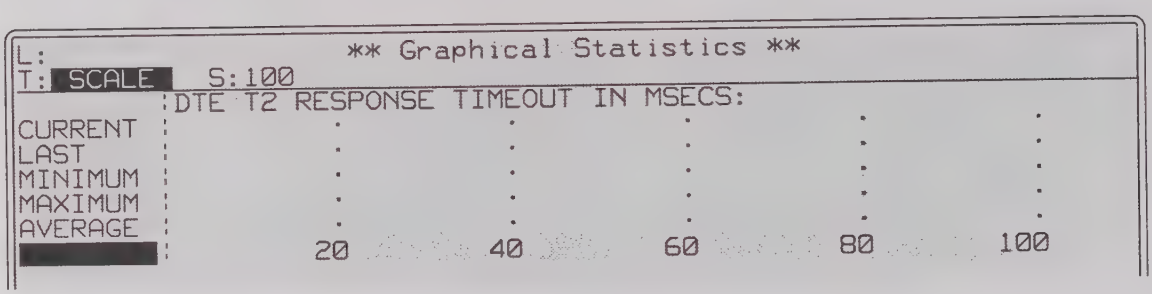


Figure 22-6 The scale line at the bottom of the figure was created by the menu selections at the top.

(C) Name

The **N** field appears when you have selected COUNTER, TIMER, or ACCUM in the **T**(ype) field. Enter the name of a counter, timer, or accumulator that you have created in your program. In Figure 22-6, the labels all pertain to values called out under the name **t2**.

(D) Value

Counters, timers, and accumulators have a set of statistical values associated with them. Any of these values can be represented by a bar on the graphics display. Select one of these values in the **V** field. Selections under **V** are AVG, CRNT, LAST, MIN, and MAX. In Figure 22-6, each label references a different value of timer t2.

Note that accumulators do not have a current value: see Section 21.4.

(E) Scale

Each bar line is 54 columns wide. The scale field allows you to pick a number that will display a bar that is *fifty* columns wide. When the statistical value you have selected for graphic display attains this number, its bar will *almost* fill the width of the line.

The **S**(cale) entry for a counter or timer value has no direct relation to any scale *line* (see Type, above) that may be drawn above or below it. The scale line merely writes numbers on the screen. The **S** selection for a counter, timer, or accumulator will scale the actual bar to your estimate of what the maximum value will be. If your estimate is good, the bar will have some magnitude without overflowing the width of the screen.

If your bar is drawn to too small a scale and it overflows, go back to the Graphical Statistics screen in Program mode and increase the **S** value for the counter or timer. The statistical values are kept during Program mode (until you hit RUN again), and the bar will be redrawn to the new scale as soon as you move the cursor up or down.

(F) Intensity

Three degrees of intensity are selectable for any horizontal bar. In the **I** field, select 100% for full intensity (white against a dark background), 50% for half intensity (medium gray), and 33% for low intensity (light gray). All three bar intensities are shown in the Run–mode graphic displays in Figure 22-7. (Remember that whites and blacks are inverted in the screen illustrations in this manual.)

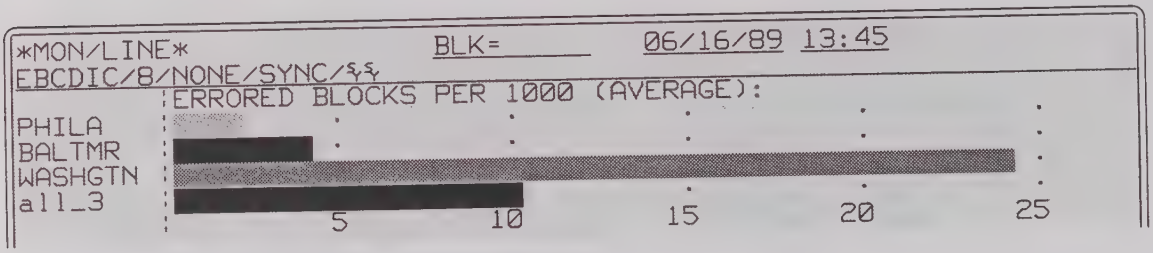
(G) Color

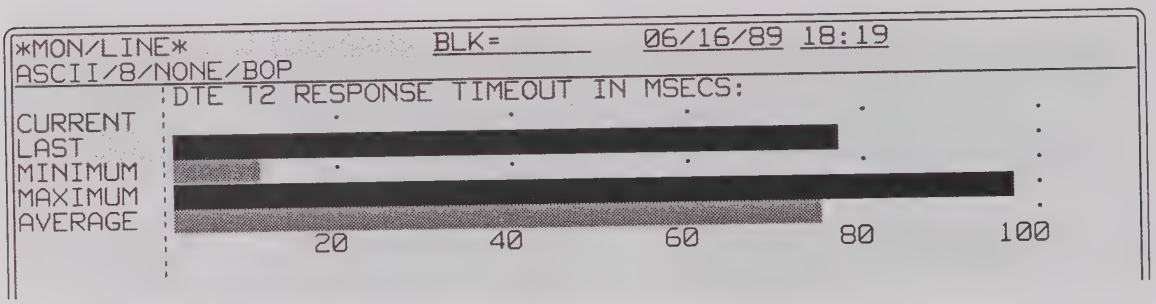
Selections in the **C**(olor) field are WHITE, RED, GREEN, BLUE and YELLOW. The selection in this field has no effect on the screen of the INTERVIEW, but it does affect the signal transmitted on the RGB interface at the rear of the unit. If a color monitor is attached at this interface, horizontal bars on the color–graphics display will be white, red, green, blue or yellow, according to the color selected for each bar (subject to the intensity selected for that bar).

(H) Unit

The **U**(nit) field appears whenever TIMER is the Type selected. Selections in this field are SECS, MSECS, and uSECS. The scale numbers on timing graphs relate directly to these units.

Do not select a timer unit that is smaller than the tick interval (**Tick Rate**) selected on the Front–End Buffer Setup menu. This rule of thumb is explained in Section 9.1(C), Time Ticks in Relation to Timer Units.





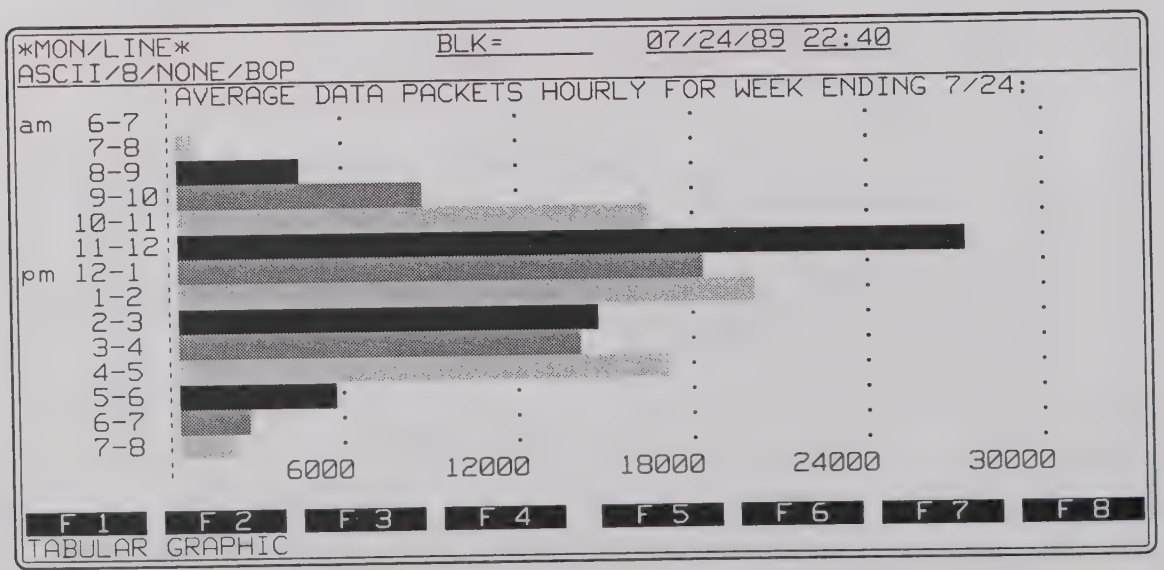


Figure 22-7 Three graphic displays.

23 Three–Tiered Programming

The INTERVIEW 8000 Series is designed to provide programming solutions for problems of varying complexity and for users with different levels of programming skill (see Figure 23-1). The simplest programming tool is the Trigger Setup screen. The setup screen guides the user through a fundamental set of programming selections.

The Protocol Spreadsheet is a more sophisticated and flexible programming method. While based on the same principles as the Trigger Setup screens, the Protocol Spreadsheet provides free–form programming options and a more advanced set of conditions and actions. The spreadsheet allows branching from program routine to program routine as well as simultaneous testing for different sets of conditions. In addition, the structure of the Protocol Spreadsheet is modeled after OSI layered architecture described in Section 24.

A third programming method is present in the INTERVIEW 8000 Series: C programming language, accessible from the spreadsheet, allows the advanced user to write code for test situations outside the scope of standard spreadsheet test selections.

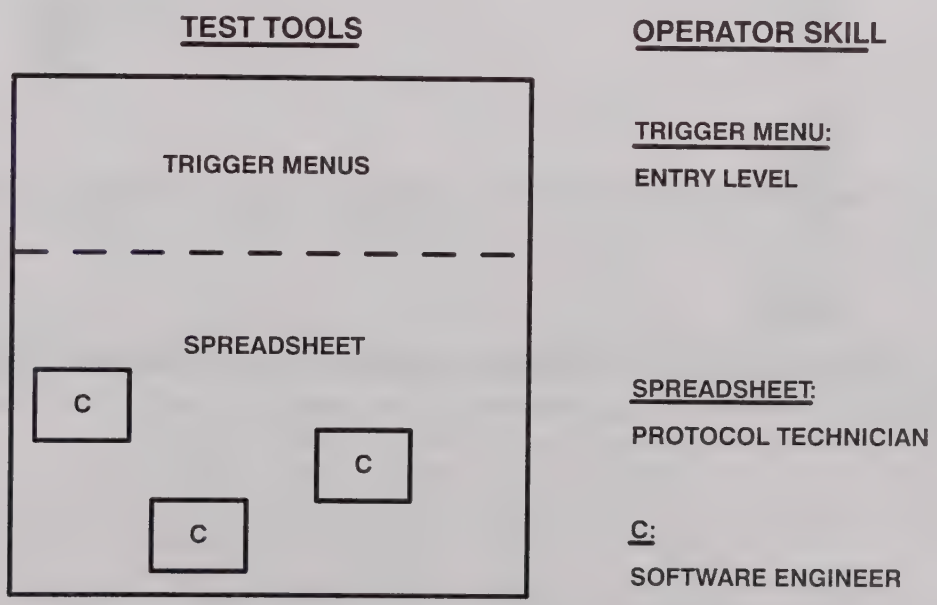


Figure 23-1 There are three separate, *integrated* user–interfaces for programmers of the INTERVIEW 8000 Series.

23.1 Trigger Setup Screens

Triggers are the basic programming tools behind all of the INTERVIEW's test activity. The operator creates each trigger in one of two ways: by using a preexistent Trigger Setup screen or by keying in trigger conditions and actions on the INTERVIEW's Protocol Spreadsheet.

A trigger is a distinct set of conditions (input) and actions (output). That is, a trigger waits for a specified event or group of events. (These events might include, for example, receipt of a certain data string or change in an internal counter.) When all conditions are met, the trigger responds with a specified action or group of actions. (Trigger actions might include transmission of a data string, sounding of an alarm, or setting of an internal flag.)

There are 16 Trigger Setup screens available in the INTERVIEW 8000 Series. A sample Trigger Setup screen is shown in Figure 23-2. These preconfigured screens provide a simplified approach to programming. Possible conditions are grouped at the top of the menu, and actions potentially taken in response to those conditions are grouped at the bottom of the menu. Trigger conditions contained on the Trigger Setup screens are described in Section 25; Trigger Setup actions are described in Section 26.

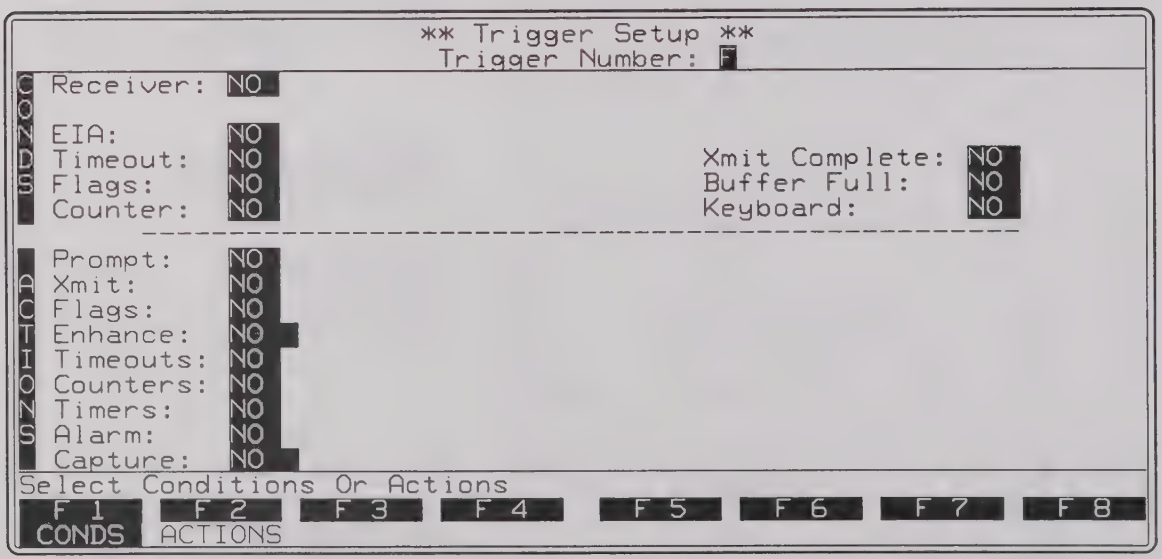


Figure 23-2 There are 16 predefined Trigger Setup screens in the INTERVIEW 8000 Series.

23.2 The Protocol Spreadsheet

The Protocol Spreadsheet, while not a prefabricated menu, contains and extends the set of programming options available on the Trigger Setup screens. As explained in Section 24, the Protocol Spreadsheet program is divided into layers, which are in turn subdivided into smaller components. At each layer, a different protocol is applicable. Depending on the protocol packages which you load, the set of trigger conditions and actions is enlarged to include automatic selections tailored to the protocol and layer you are programming. (Protocol packages are loaded from the Layer Setup screen as described in Section 8.) Figure 23-3 shows the beginning of a spreadsheet test.

Trigger conditions and actions which are always available on the Protocol Spreadsheet are described in Section 31. Conditions and actions available at Layer 1 are discussed in Section 32. Primitives used at different layers are discussed in detail in Section 34.

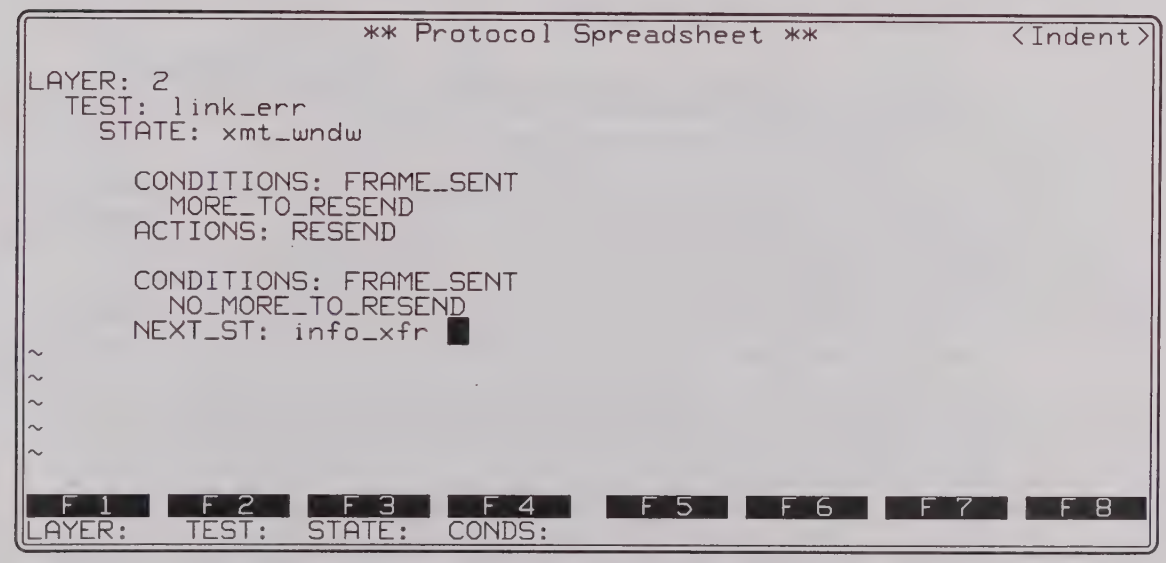


Figure 23-3 The Protocol Spreadsheet conditions and actions shown here are part of the X.25 personality package loaded in at Layer 2.

(A) Automatic Protocol Selections: Personality Packages

Standard data units for a protocol which has been loaded from the Layer Setup screen are available by name so that it is not necessary to enter long hexadecimal strings as conditions or transmit actions. You are also spared the calculation of sequence numbers, poll bits, parity, block checking, and certain other variables that must be included in a received or transmitted string.

Timeouts, window sizes, calling sequences, transmission paths, and other protocol–specific parameters can be modified on a submenu which accompanies the personality package.

Protocol–specific conditions and actions are discussed at the end of this manual in a section devoted to the protocol and layer.

(B) Creating and Editing Spreadsheet Programs

Protocol Spreadsheet triggers are created by the operator through the use of indexed softkeys. The entries you make become visible on the screen only after you have made a selection. You also have the option of typing your program from the regular keyboard, as long as keyed entries match the text keywords which are displayed on the screen once you press the function keys. Syntax errors are indicated by a strike–through as you type or make function–key entries.

Press [EDIT] to invoke an alternate bank of spreadsheet keys which provide advanced editing functions. All editing functions are described in Section 30.

23.3 C Programming Language

The INTERVIEW version of C is based on the current ANSI recommendations for C programming language, with extensions to provide multitasking. C is intended as an aid to users who have advanced programming knowledge.

C statements can be incorporated in the spreadsheet as conditions or actions. Figure 23-4 shows C included as a trigger action which displays a prompt at the top of the screen and incorporates a counter value as part of the message. This gives you the ability to extend existing spreadsheet selections or to construct an entire test from scratch using C. C allows you the freedom, for example, to create a customized protocol or program trace display or to manipulate variable data strings anticipated within a user–specific protocol.

```
                    ** Protocol Spreadsheet**

  STATE: bad_fcs
    CONDITIONS: DTE BAD_BCC
    ACTIONS: COUNTER badfcs INC
    {
     pos_cursor (0,0);
     displayf ("DTE bad frames: %ld",
                          counter_badfcs.current);
    }
~
```

Figure 23-4 The *displayf* function in a C window allows you to write a variable such as a counter to the top of the data screen during Run mode.

23.4 Integrating Programming Methods

The three tiers of programming, Trigger Setup screens, Protocol Spreadsheet, and C programming language, can be integrated to match the needs of each user. The 16 preconfigured Trigger Setup screens can, for example, be employed as a simple

line–monitoring test operating at Layer 1. The Protocol Spreadsheet program can later add more complex tests to this, so that several tests are operating simultaneously at a number of layers (see Section 24 for a discussion of layered architecture). Within the Protocol Spreadsheet program, unique test situations or test problems of particular complexity can be resolved by including C programming statements.

(A) Variables Shared Between Trigger Menus and the Spreadsheet

Certain internal program controls are shared between spreadsheet and Trigger Setup screens, in order to allow communication and interdependency between the two types of testing. Internal counters and program timeouts which have the same name can be monitored and controlled both from trigger screens and from the spreadsheet.

There is limited sharing of internal flag bits between Trigger Setup screens and the Protocol Spreadsheet. Trigger Setup flag bits are shared between all trigger screens. They can also be monitored or changed on the spreadsheet, where they are referenced as a flag named **trig_flag**.

(B) Saving and Loading Program Segments

The INTERVIEW's filing system provides a means for storing entire programs or portions of programs for later use by operators of any skill level. On the File Maintenance Screen, you may specify which group of menus you wish to save. For example, if you specify a "Setup," only the five setup menus (Line Setup, Interface Control, BCC Control, Front–End Buffer Setup, and BERT Setup) are saved. This allows you the freedom to create new trigger or spreadsheet tests without continually reconfiguring all menus.

If you save a "Program" on the File Maintenance screen, you are saving the configuration of all menus, including Trigger Setup screens, Layer Setup, and the Protocol Spreadsheet. The one menu that is not saved is the Printer Setup. A program may be a simple test ready to be enlarged, or it may be a highly complex group of tests that can be loaded and run by an operator with little or no programming knowledge.

As a complement to file maintenance options, the Protocol Spreadsheet editor allows you to save only the spreadsheet portion of a program. Advanced programmers can create a set of tests on the spreadsheet or text files containing C code, then use the editor to write their work to a file. Later, operators may load a setup or a partial program from the File Maintenance screen, call up the spreadsheet screen, and use the editor to read in the advanced programmer's file in order to complete their own program.

> **NOTE:** The File Maintenance Compile command also can be used to save the contents of the Protocol Spreadsheet. The linkable–object file which results contains the compiled object–code version of the program. See Sections 15.3(P) and 28.4.

24 The Layered Program Model

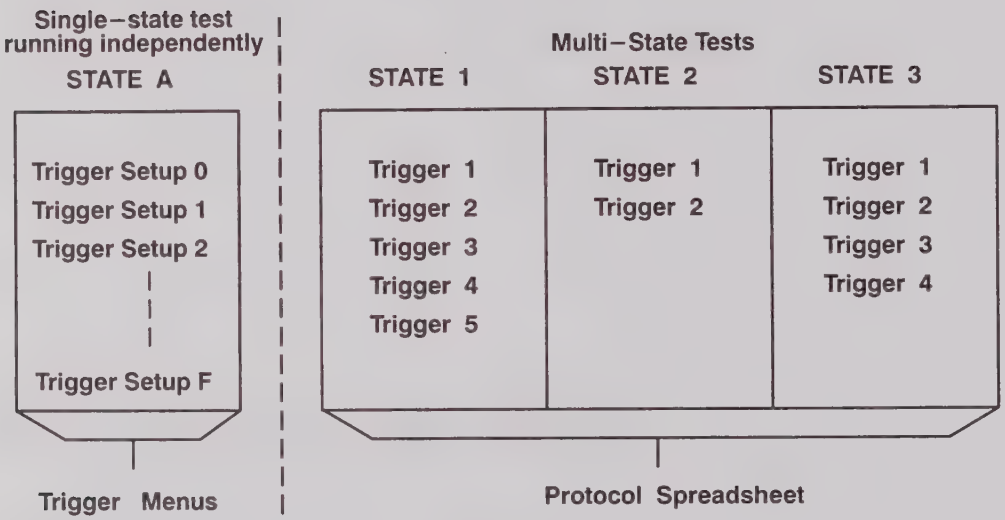


Figure 24-1 Triggers on the Protocol Spreadsheet are grouped into States which can be called in varying order.

24 The Layered Program Model

The trigger, described at the beginning of the previous section, is the fundamental component of all INTERVIEW programs. On the Protocol Spreadsheet, the trigger is grouped to form larger programming blocks, referred to as states. States are grouped to form tests. And tests are divided into layers. The largest component of the INTERVIEW program, the layer, is patterned after the Open Systems Interconnection (OSI) model.

24.1 States

It is a useful programming procedure to group triggers so that some are inactive while others are active. This is possible to a limited extent on Trigger Setup screens, using counters, timeouts, or internal flags to sequence the triggers.

On the Protocol Spreadsheet, triggers can be more easily grouped by separating them into States (Figure 24-1). A state is an independent group of simultaneously active triggers called into play as required by the test. Within a test, only one state is active at one time. That is, all the triggers in this state are awaiting input. All other triggers in the test are dormant.

(When Trigger Setup screens are compiled into the Protocol Spreadsheet, they may be thought of as a single–state test constantly running at the Layer 1 interface.)

When one of its triggers receives the right input, the active state passes control to another state and itself becomes inactive. Transitions between states are controlled by (spreadsheet) triggers. These trigger–controlled transitions between active and inactive states make branching possible. That is, a more complex set of conditions can be set up inside a state and certain actions can ensue. Then, at a decision point (for example, "Did the receiving party respond to my transmission?"), the test can choose the correct path from several potential paths. (A: "Yes, they answered, so transmit next message"; or B: "No, they didn't answer, so resend previous message.")

24.2 Tests

Even further flexibility is possible in INTERVIEW programs, because states can be grouped into tests. So, not only does the INTERVIEW move back and forth laterally between the groups of triggers contained in various states, but it can also use different sets of states to perform several different tests at the same time (Figure 24-2). As an example, two simultaneous tests might be used to check the different set of exchanges expected on either side of a full–duplex line.

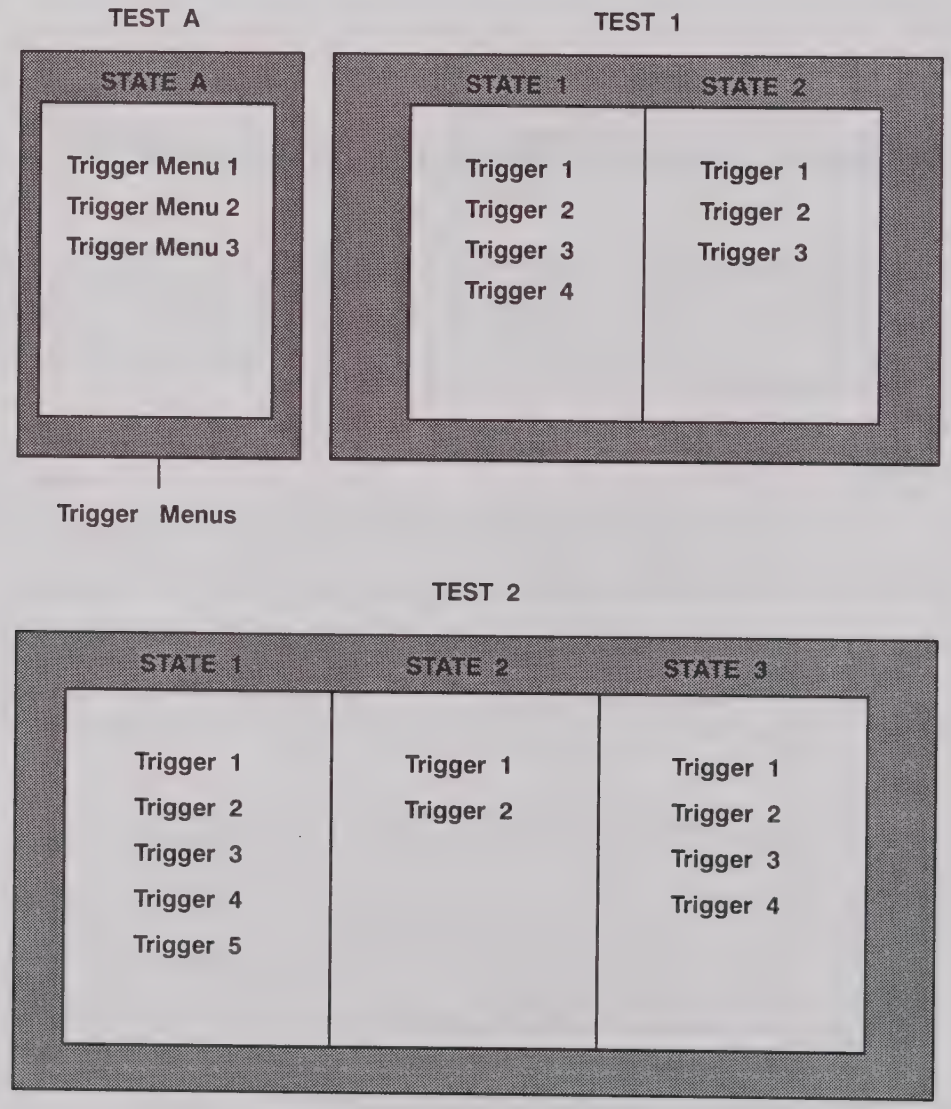


Figure 24-2 Distinct sets of states can be created so that the INTERVIEW can perform several different tests at the same time.

24.3 Layers and the OSI Model

Finally, groups of simultaneous tests can be "layered." In this way, separate tests or groups of tests can be run at a maximum of seven levels simultaneously (Figure 24-3).

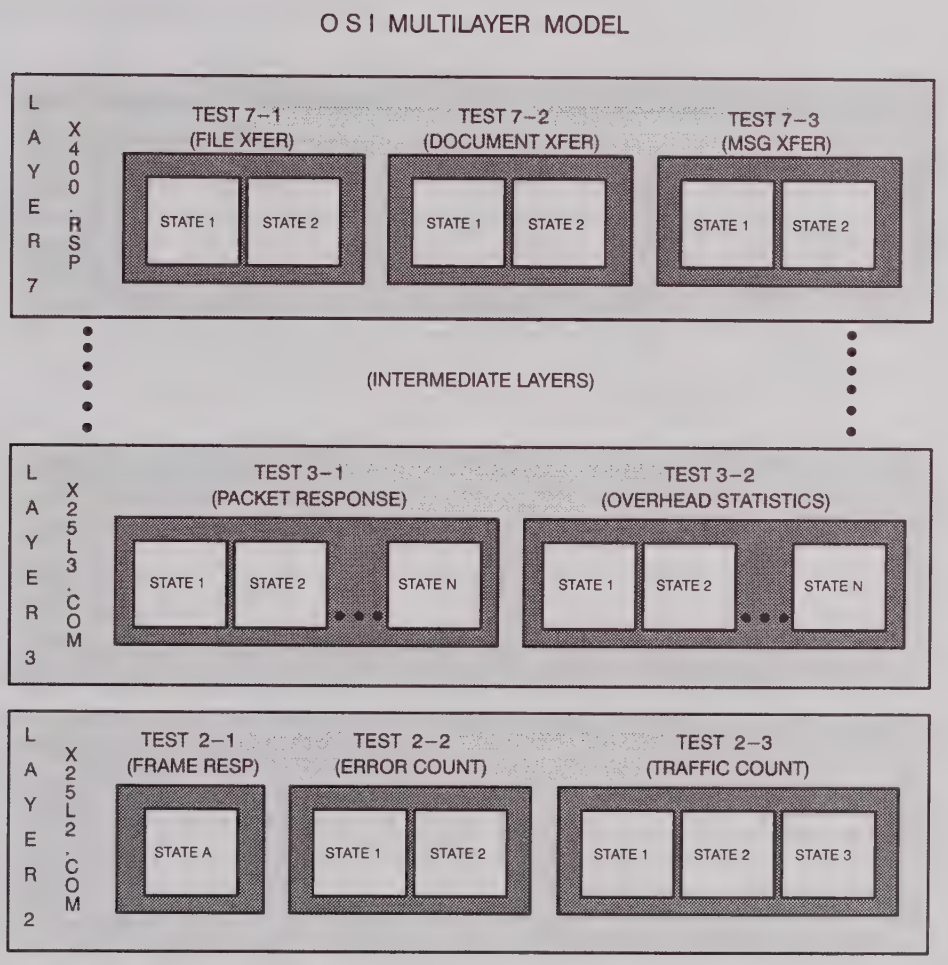
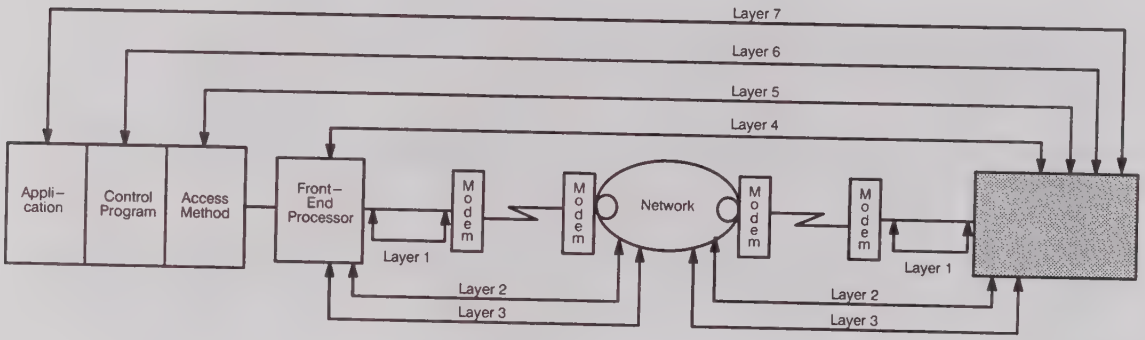


Figure 24-3 Separate tests or groups of tests can be run at a maximum of seven layers simultaneously. This capability parallels the OSI seven–layer model.

This layered structure is specifically designed to handle protocols which conform to the CCITT Open Systems Interconnection (OSI) model. The OSI model is fully described in CCITT Recommendation X.200.

This is a seven–layer model (see Figure 24-4) in which each layer performs a different data communication function. Conceptually, each layer is independent. One layer can be modified without other layers being affected, as long as the modified layer respects prescribed communication with the layer immediately above and the layer immediately below it.



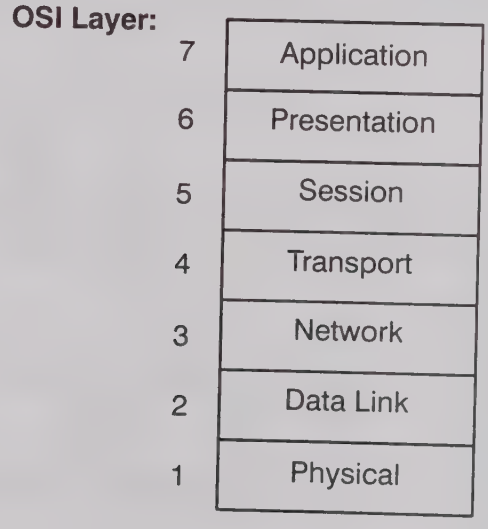


Figure 24-4 The seven OSI layers.

Suppose that the physical link between two nodes in a network were changed from copper wire to optic fiber. In an OSI configuration, only the physical layer (and possibly certain aspects of the data link layer) would be modified. The remainder of the communication process would stay the same.

The separation of programs into discrete layers generally reduces the complexity of test conditions and actions. This simplifies programming for the user. The structure allows you to verify your system—and to debug your own tests—layer–by–layer. For example, it is not necessary at Layer 3 of a protocol to anticipate variations in line–level or frame–level events. Searches for strings and protocol elements focus only on the portion of a frame which pertains to Layer 3. The validity of the frame which contains the string has already been checked.

24.4 Personality Packages

The layered structure of the OSI model allows you to use different protocols at different layers—again, provided that the rules of OSI interlayer communication are observed.

The INTERVIEW provides layer–specific protocol packages, called personality packages, which you can load from the Layer Setup screen. While certain layer protocols are more commonly used together (SDLC at Layer 2 and SNA at upper layers, for instance), it is possible to mix and match them. You could, with the correct Personality Package, load and run X.25 protocol at Layer 2 and SNA protocol at higher layers.

Personality packages are not in themselves protocol emulations; rather, they are high–level interfaces to routines in the given protocol. A package at Layer 2 X.25, for example, allows users to *design their own applications* by simple softkey–entry of a routine such as:

```
CONDITIONS: RCV DISC
ACTIONS: SEND SABM
```

or

```
CONDITIONS: T1_EXPIRED
ACTIONS: RESEND
```

Personality packages are selected and loaded from the Layer Setup screen (shown in Figure 24-5; see Section 8 for a description of this screen). The contents of each personality package are described in a section dedicated to the package (refer to the Table of Contents, Sections 36, and following).

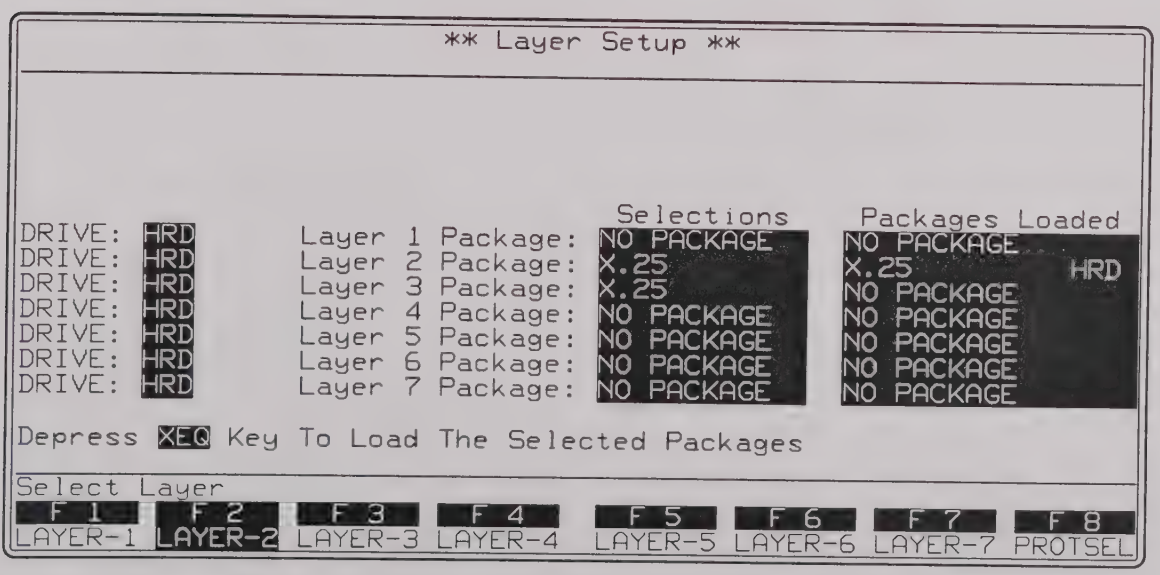


Figure 24-5 Personality packages, which provide the protocol elements in INTERVIEW programming, are loaded from the Layer Setup screen.

24.5 Primitives

The OSI Layers use limited–range messages called *primitives* to communicate with each other. Primitives are defined by the OSI model and are not linked to any one protocol. No matter what personality package is loaded, these generic primitives are available at each layer. This gives you the freedom to create or modify a protocol. Primitives available on the Protocol Spreadsheet are discussed in Section 34.

24.6 Constants

To represent a frequently used test value, you may define a constant once in your program and reference the constant elsewhere in the program as needed. Replacing the test value with a new value then becomes easy, since you need only change the constant definition one time.

Constants, which may be used to represent any textual string, can be defined at several levels in the spreadsheet program. The function key labeled CONSTS: is only present when it is legal to define constants.

Depending on where they are defined, constants vary in scope. You have the option of creating constants which can be used globally, throughout a layer, or throughout a test. Refer to Section 29 for a full description of constants.

25 Trigger Conditions

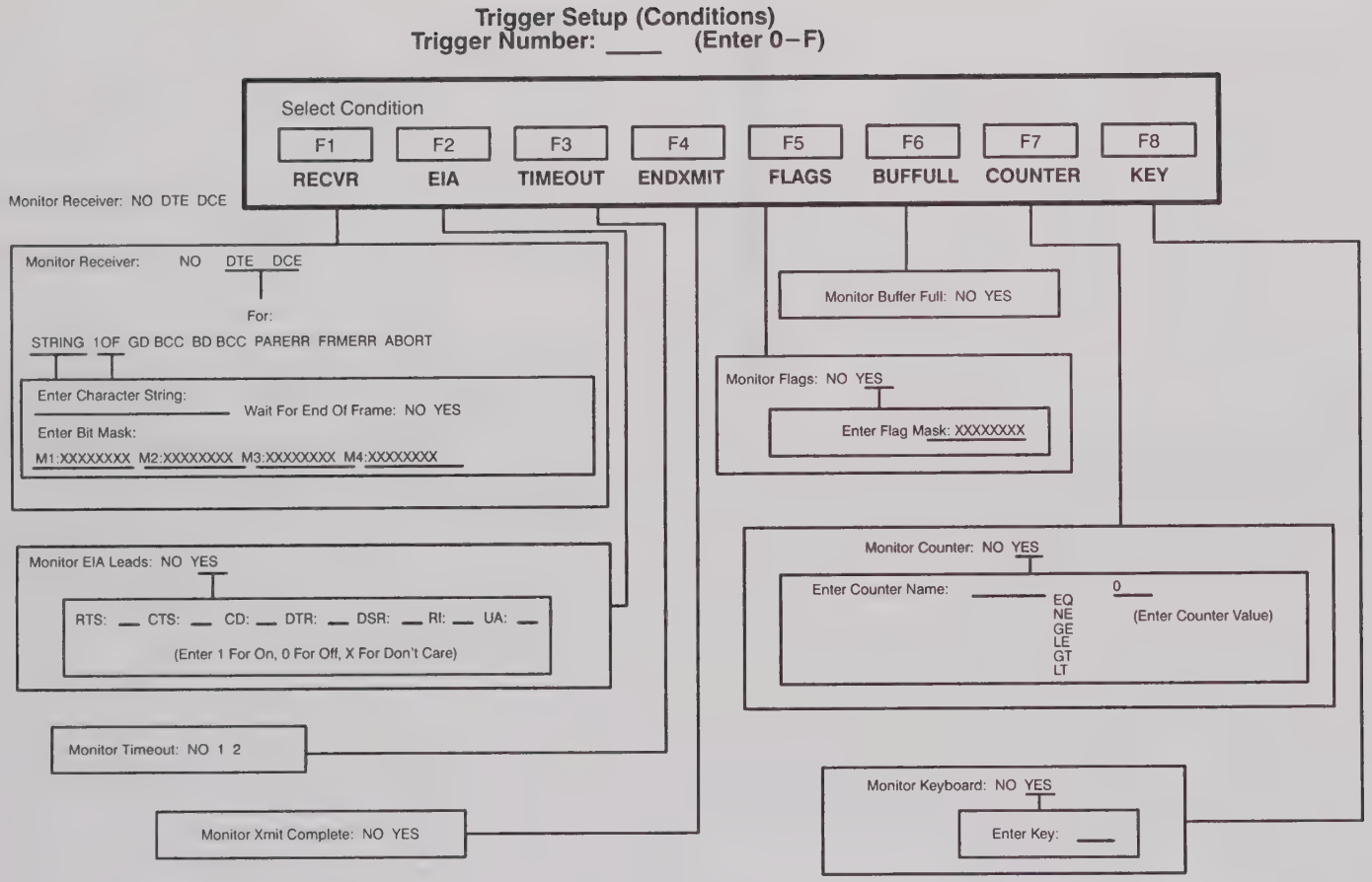


Figure 25-1 Conditions on Trigger Setup menu.

25 Trigger Conditions

Triggers can be thought of as "IF, THEN" statements, represented on the screen as "Conditions" (IF...) and "Actions" (THEN...). This section pertains to Trigger Conditions available on the preconfigured Trigger Setup screens, of which there are 16 in the INTERVIEW 8000 Series. All possible conditions available on the Trigger Setup screen are shown in Figure 25-1.

Triggers are numbered 0 through F. To access a particular trigger screen, press the TRIGS function key on the Main Program Menu. This calls up the Trigger Summary screen. Enter the Trigger Number desired (it will appear highlighted at the top of the screen) to see that trigger screen.

Each trigger screen is divided in half, with Conditions at the top of the screen and Actions at the bottom of the screen. A default Trigger Setup screen is shown in Figure 25-2.

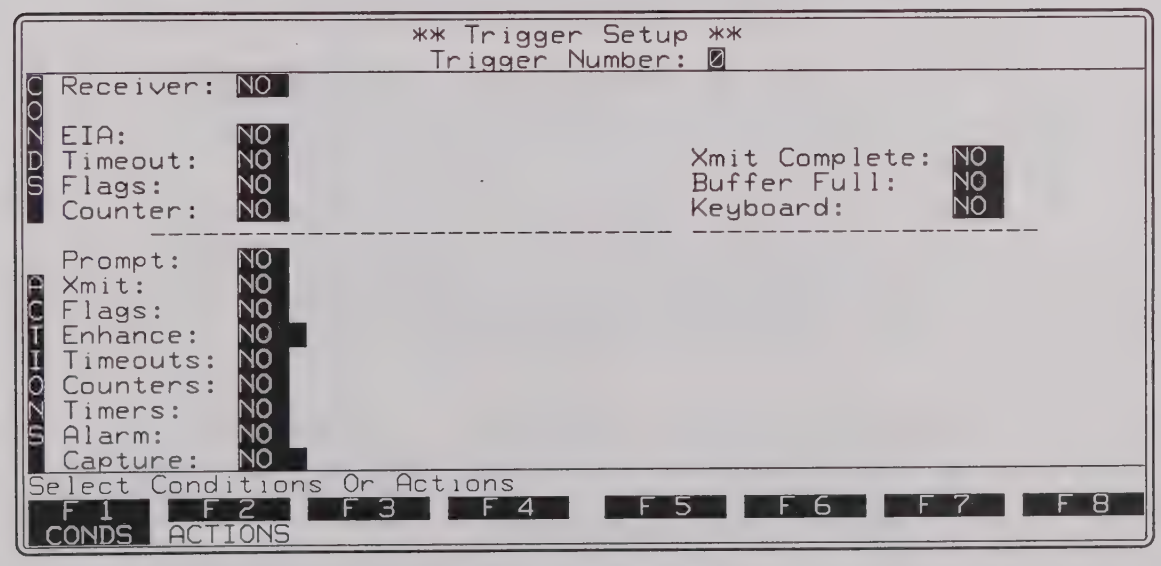


Figure 25-2 Default trigger menu.

25.1 Active Triggers

Only active triggers are tested. Trigger Setup screens are continuously active. (This is not true of triggers on the Protocol Spreadsheet, where triggers are configured in alternately active states as a matter of program design.)

25.2 Combining Conditions on the Same Trigger Setup Screen

A trigger is true and can take action only at the instant that *all* trigger conditions are met.

(A) Static vs. Instantaneous Conditions

Internal flag, Counter, EIA lead, and Buffer Full conditions differ from other conditions on the Trigger Setup screen. When it is used in a trigger *by itself*, each of these conditions, like other independent conditions, initiates its trigger actions only at the instant that it transitions to true. In addition, these four conditions can retain a *status* of true for a long period of time.

The *static* value of these four conditions is tested for true or false when they are combined in the same trigger with another condition.

All other trigger conditions are true only at the instant that they happen. We will refer to them as "instantaneous" or "transitional" conditions.

> **NOTE:** It is important to remember that even a "static" condition is "transitional" when it is used *alone* in a trigger. An EIA condition, for example, used by itself in a trigger cannot come true without a transition.
>
> An exception to this rule is when the test enters Run mode. At that moment, static conditions—flags, counters, EIA leads, and buffer full—used alone on a Trigger Setup menu (*not* on the Protocol Spreadsheet) are tested once for a status of true. Then they revert to being true only upon transitions.

(B) Rules for Combining Conditions

These "static" conditions can be combined with other trigger conditions on the same Trigger Setup screen to form compound "IF" statements. Here are some rules to remember in combining trigger conditions:

1. When "static" conditions appear on the same trigger menu with an "instantaneous" condition, the trigger is keyed to the instantaneous condition. All static conditions must be true when the instantaneous condition transitions to true. On that transition, trigger actions are taken.

   Suppose, for example, that a trigger is looking for a Bad BCC and a counter value = 20. The counter value must first increment to 20, then the Bad BCC

must be detected. As soon as the Bad BCC is detected, the trigger becomes true and takes action.

2. When static conditions are combined, both (or all) are transitional. When one of them transitions true, the other(s) becomes a static condition and is checked for a status of true. The user does not have to try to anticipate which of two (or more) conditions will transition first.

> **NOTE:** On the Protocol Spreadsheet, static conditions are prioritized in the order that the user lists them: only the first is transitional. The Protocol Spreadsheet therefore requires you to define which static condition will be the controlling, transitional condition. See Section 31.2.

25.3 Receiver

This condition monitors the data lead specified (DCE or DTE) for designated data. When **Receiver:** DCE or DTE is selected, several options become available: String, 1of, Good BCC, Bad BCC, Parity Error, Frame Error, and Abort.

(A) DTE or DCE

In using the Receive condition, you must specify which side of the line you wish to monitor. Select DTE to denote the TD lead. Select DCE to denote the RD lead.

(B) String

This selection allows you to enter a string of up to 16 characters in the field provided. The entire, exact sequence of characters entered must be received for the condition to be true.

(C) "One of"

When 1 OF is selected, the trigger looks for any one of the characters entered in the next field. Up to 16 individual characters can be entered.

(D) Good or Bad BCC

GD BCC (Good Block Check Calculation) and BD BCC (Bad Block Check Calculation) cannot be used as conditions unless **Rcv Blk Chk** is on in the unit (selectable on the Line Setup screen; see Section 5). Select GD BCC or BD BCC when you want the trigger to take action on receipt of the Block Check Calculation (referred to as FCS, or Frame Check Sequence in Bit–Oriented Protocols).

(E) Parity Error

PARERR looks for a parity error in relation to the **Parity** selection made on the Line Setup screen.

(F) Framing Error

FRMERR applies to start–stop formats (ASYNC and ISOC) and locates framing errors, based on the stop bits anticipated. Both **Format** and **Stop Bits** are selected on the Line Setup screen.

(G) Abort

This selection applies to all Bit–Oriented Protocols. When ABORT is selected, the INTERVIEW triggers off of the seven consecutive 1–bits which constitute an Abort. BOP should be used as **Format** on the Line Setup Menu when ABORT is selected.

> **NOTE:** The trigger condition will not respond to idle–time aborts, unless **Display Abort:** YES has been selected on the Line Setup screen.

(H) Character Entry Field

This field appears only if STRING or 1 OF is selected. It is the data–entry line for a sequential character string, if STRING has been selected; or a nonsequential character list, if 1 OF has been selected. Up to 16 characters may be entered, in either case.

1. *String entry*. The 16 characters allowed in the string may include any of the following in any order or number:

   All upper and lower–case ASCII characters available on the keyboard.

   All control character mnemonics on the keyboard.

   Two–digit hexadecimal entries. These are entered by first turning the HEX key on, then using alphanumeric keys 0 through 9 and A through F. Two alphanumeric key strokes are required for each hex character. A hex character is represented on the screen as a pair of small characters, the first ascending and the second descending. Compare hex characters to regular alphanumeric characters in Figure 25-3.

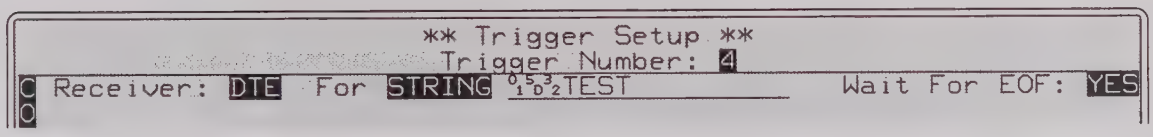


Figure 25-3 Both alphanumeric and hexadecimal characters can be entered as part of a condition search string.

Characters entered in hexadecimal are not translated, and parity is not calculated for them; therefore, you must include the parity bit, whether good or bad, in your entry.

2. *Flags*. You must press [FLAG] to enter the ᵀE Flag byte used in Bit–Oriented Protocols. The INTERVIEW's logic will not read a hexadecimal entry made with [HEX] as a flag.

3. *Sync*. Press [SHIFT]–[FLAG] to enter the sync symbol. The character [S] is displayed on the Trigger screen.

4. *Not equal (≠) entry*. When a character key is preceded by [NOT EQUAL], all characters not equal to that character will satisfy that position in the string. These characters are represented in the data entry field with a horizontal bar through them.

5. *Don't care*. [DON'T CARE] permits *any* character received in that position to satisfy the condition.

6. *Bit masks*. Four bit masks can be positioned anywhere in the data–entry string. To enter a bit mask, use [BIT MASK] at the desired location in the string. Each time you press [BIT MASK], a new mask field appears below the string–entry field. To move the cursor to the next position in the string–entry field, press [DONE]. The mask fields are numbered M1 through M4, to denote the order in which they appear in the string (see Figure 25-4).

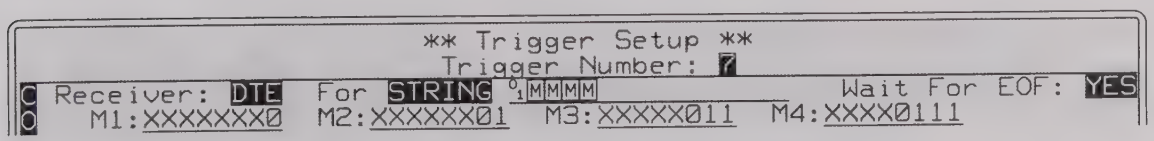


Figure 25-4 Four bit masks can be used as part of the search string.

If you have used Bit Masks 1 and 2, for example, at positions 5 and 8 of the string (as in Figure 25-5), you may decide to change the character at position 1 to a bit mask. Return the cursor to position 1 of the string and press [BIT MASK]. The character

at position 1 will be overwritten with M, the two prior bit masks will be renumbered to 2 and 3, and their menu location shifted to make room for the new Bit Mask 1. The cursor will be on the left–most bit of the new mask. (Compare Figure 25-5 and Figure 25-6.)

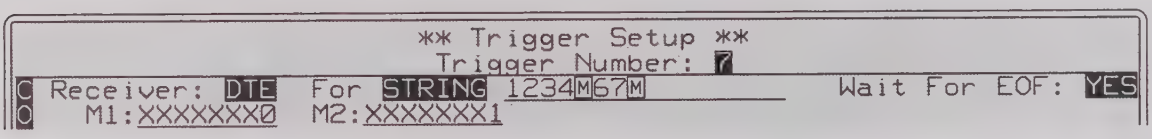


Figure 25-5 Bit masks M1 and M2 are entered at positions 5 and 8 of this string.

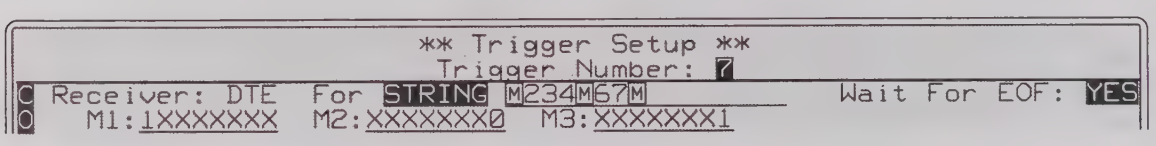


Figure 25-6 A third bit mask has been entered at position 1, and the old M1 and M2 have shifted automatically to M2 and M3.

NOTE: When a Bit–Oriented Protocol is being tested, the INTERVIEW ignores inserted zero bits. You can specify the search string on the Trigger Setup screen without considering zero bit insertion.

7. *Double parens.* A global constant declared on the Protocol Spreadsheet can be entered as part of a Receiver String. Enter the name of the constant exactly as it appears on the spreadsheet and enclose it in double parentheses, as follows: ((drop)). The double parentheses are special characters created by pressing CTRL–9 for ((, and CTRL–0 for)). When constants are used, the receive string cannot be longer than 32 characters *after all constants are expanded*.

(l) "1 OF" Character Entry

Up to 16 characters may be entered. The same types of characters valid in **String** entry are valid **1 OF** characters. The trigger will take action upon receipt of the first character that matches any one of the characters anywhere in the list. If DON'T CARE is used, only this character should appear in the **1 OF** entry field (since any character is a match for Don't Care). If *one or more* characters in the field are entered as NOT EQUAL, only those characters not in the field will satisfy the condition. Thus, ~pqz in the **1 OF** field means the same thing as ~pqz: all characters other than p, q, and z will satisfy the condition.

(J) Wait For End of Frame

This is a subfield which appears when STRING or 1 OF is selected. The default selection is NO. When **Wait For EOF:** YES is selected, the trigger first tests for the data specified on the trigger. Then it evaluates the block check at the end of the frame. The trigger will not take action when a Bad Block Check or an Abort is detected on a received frame.

25.4 EIA

Select YES in the **EIA** field if you want a trigger to monitor status of up to seven RS−232/V.24 leads (see Figure 25-7).

> **NOTE:** For line data, EIA lead−status is not detected if control leads are not buffered in the Front−End Buffer. See Section 9. For recorded data, EIA lead−status is not detected if control leads were not buffered in the FEB at the time of recording.

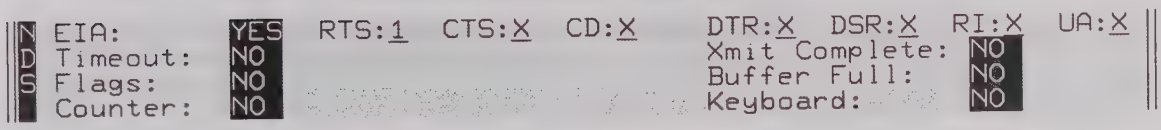


Figure 25-7 Each trigger can monitor the status of seven EIA leads.

Enter a 1 in the box under a lead to indicate ON; a 0 for OFF. No entry (X) is read as Don't Care. Entry fields are provided for six leads: RTS (Pin 4), CTS (Pin 5), DSR (Pin 6), DTR (Pin 20), CD (Pin 8), and RI (Pin 22). You may monitor a seventh RS−232/V.24 lead by strapping the desired lead to UA on the Test Interface Module (see Section 12 for instructions).

In using the **EIA** condition, you should keep the following points in mind.

- If only **EIA** conditions are selected, the trigger will wait for all **EIA** conditions to be satisfied. It will become true on the last transition necessary to satisfy these conditions.
- The **EIA** condition is a static condition. The rules for combining static conditions with other conditions are explained in Section 25.2.

25.5 Timeout

Two timeout timers can be monitored from the Trigger Setup screen.

The decrementing timeout timer is set for a specific time as part of a trigger action. The trigger which sets a monitored timeout may be any of the Trigger Setup screens or any trigger on the Protocol Spreadsheet.

The default **Timeout** selection is NO. Select 1 to monitor Timeout timer 1; 2 to monitor Timeout timer 2. The condition is satisfied at the instant of the timeout.

25.6 Transmission Complete

When **Xmit Complete:** YES is selected, the INTERVIEW tests for the end of its own transmissions.

> **NOTE:** The INTERVIEW transmits only when operating in Emulate DTE or DCE mode (selectable on the Line Setup screen; see Section 5).

The **Xmit Complete** condition is frequently used with an internal flag or counter condition to control when or how many times it will be tested.

25.7 Internal Flag Bits

There are eight internal flag bits reserved for the INTERVIEW's Trigger Setup screens. The purpose of the flag bits is to provide a simple way to interconnect several triggers in order to make one trigger dependent on another or to set up triggers in sequence. Each bit is a simple switch that one trigger may set as an action and all triggers can test later.

Internal flags may all be monitored and set by any individual trigger. (The same flag bit can be set and sensed by a single trigger.) Since each flag bit can be set and monitored separately, it can also be shared among the Trigger Setup screens.

The internal flag condition is a static condition and can be used in combination with other trigger conditions as explained in Section 25.2.

To test internal flags, select **Flags:** YES. In the flag mask which then appears, enter a 1 to test for a flag bit turned ON, a 0 to test for a flag bit turned OFF. Enter X in the appropriate position of the mask (or press DON'T CARE) if you do not wish to test a particular bit. See the example in Figure 25-8.

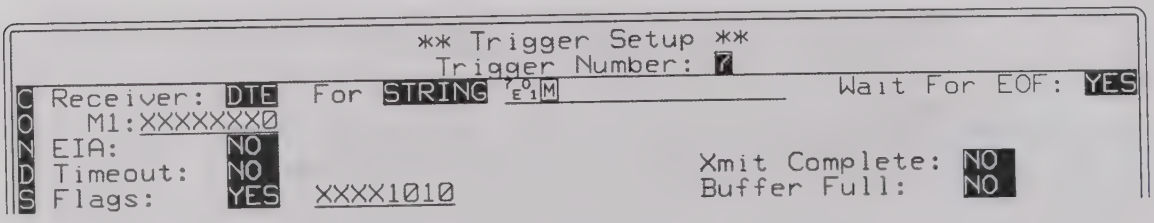


Figure 25-8 The internal flag condition is frequently used in combination with other trigger conditions.

NOTES:

All flags are set to 0 as the INTERVIEW enters Run mode.

The eight flag bits on the Trigger Setup screens are the low−order bits of a flag mask that can be accessed on the Protocol Spreadsheet by the name **trig_flag**. See Section 31.3(G).

25.8 Buffer Full

This condition checks the screen's 64 Kbyte character buffer and becomes true as soon as the buffer is full. The condition then remains true throughout the program. Buffer Full is a static condition which may be used in conjunction with other conditions. The rules for combining trigger conditions are outlined in Section 25.2.

25.9 Counter

Each Trigger Setup screen can monitor a counter with a range of 0 to 999,999. The counter which is monitored may be named and controlled either on a Trigger Setup screen or on the Protocol Spreadsheet.

The default selection is NO . When **Counter:** YES is selected, new menu fields appear (see Figure 25-9).

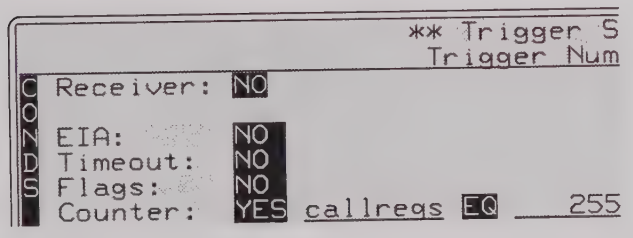


Figure 25-9 Any counter named on a trigger can be monitored as a Trigger Setup screen condition.

(A) Counter Name

Enter the counter name in the field provided. Names must start with a letter. Any of the 52 alpha characters (upper and lower) and the 10 numerals in addition to the underscore (_) character are legal in all other positions. A counter name may be up to eight characters in length.

(B) Relational Operator

Make the appropriate selection to specify when the **Counter** condition will be true. The counter may be tested for a value equal to (EQ), not equal to (NE), greater than or equal to (GE), less than or equal to (LE), strictly greater than (GT), or strictly less than (LT) the entered value on the trigger screen.

(C) Counter Value

Enter the counter value as a whole decimal number in the field provided.

25.10 Keyboard

Select **Keyboard:** YES to display a one character entry field. Then press the key which you wish to use as the condition. In Run mode when that key is pressed, the condition will be true and (if this is the only condition) will initiate a trigger action, such as a transmission. Any key *or key–combination* that produces a character listed in the ASCII chart in Appendix D1 is valid input in this field.

26 Trigger Actions

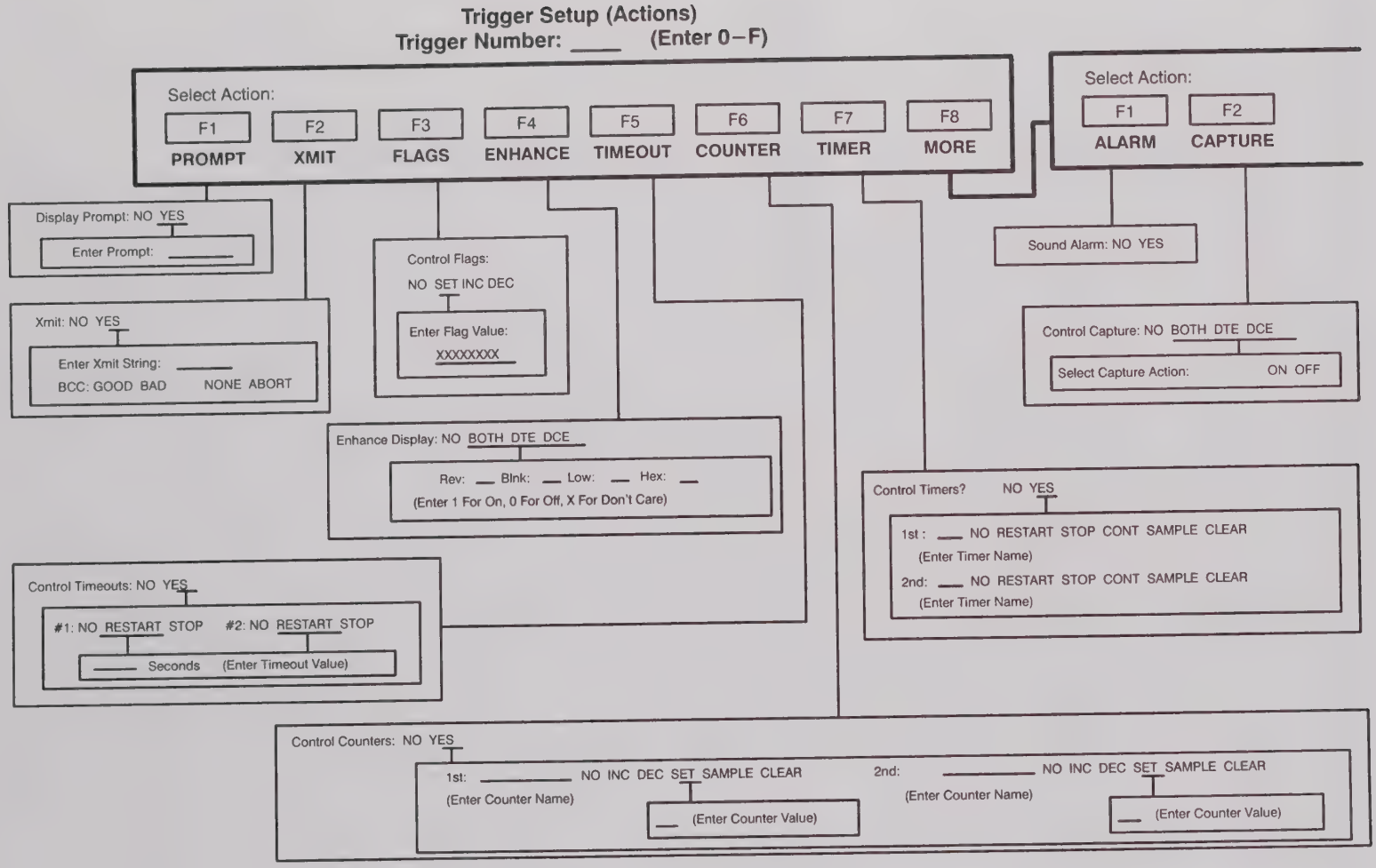


Figure 26-1 Actions on Trigger Setup menu.

26 Trigger Actions

Figure 26-1 shows all actions available on the Trigger Setup screen. Figure 26-2 shows a default trigger menu. The top half of the menu contains available trigger conditions, discussed in Section 25. A more complete set of trigger conditions and actions is available on the Protocol Spreadsheet (see Sections 31 and 32, as well as individual sections on the Protocol Packages).

All conditions selected on the top half of the Trigger Setup menu must be satisfied for the trigger to be true. Only then will the actions on the lower half of the menu be taken.

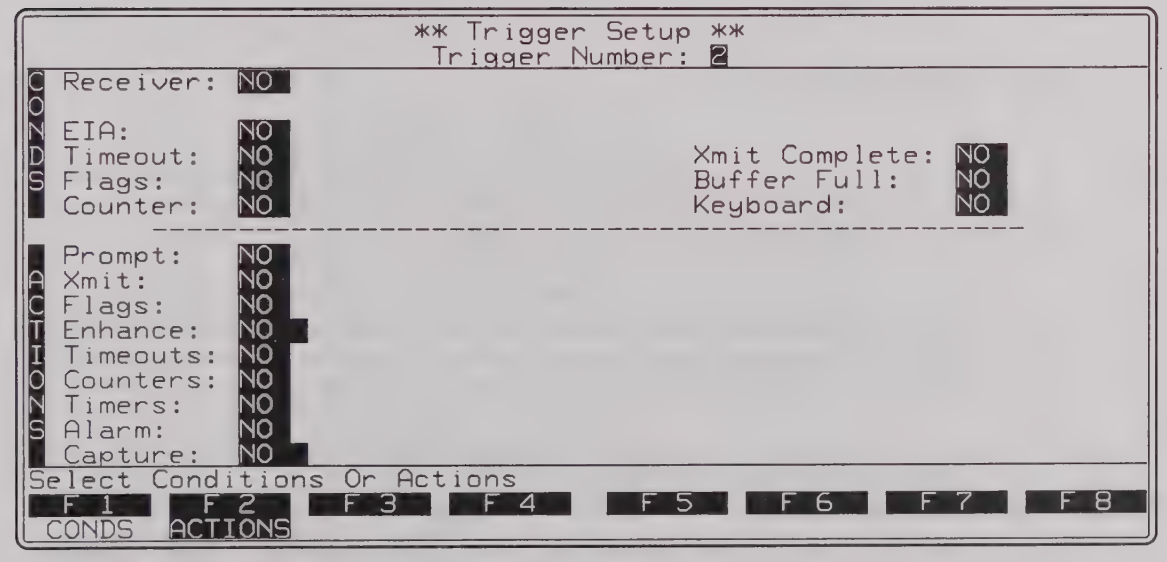


Figure 26-2 Default trigger menu.

26.1 Displaying a Prompt

Select **Prompt:** YES to display a data entry field (see Figure 26-3). You may enter a message of up to 47 characters here. Any ASCII characters are legal entries. When the trigger is true, the Prompt will be displayed on line 2 of the screen. (The prompt is NOT transmitted.) It will stay on the screen until it is replaced by another prompt, or until you clear it by changing the display mode or by pressing the RUN key.

NOTE: A new prompt does not reinitialize the prompt line on the screen. Instead, it overwrites the old prompt to the extent of the new one. If the prompt "LINK–UP" is overwritten by the prompt "CALL," the result will be "CALL–UP."

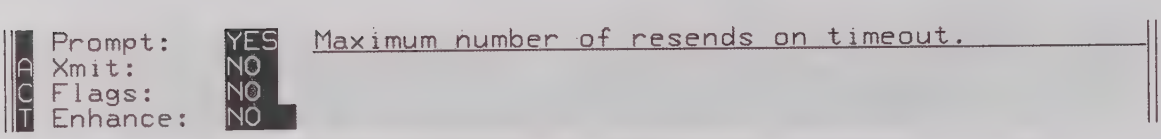


Figure 26-3 Messages entered in the *Prompt* field will appear on the second line of the screen in Run mode.

Prompts can be used to call your attention to certain occurrences, or to help you follow the course of a test. Prompts are also useful in program development because you can use a prompt to tell you when a trigger is true. Any alphanumeric or control character from the keyboard, including spaces, may be part of a prompt.

26.2 Transmitting

INTERVIEW transmissions—with the exception of BERT transmissions, described in Section 11—are continuously under trigger control, either from the Trigger Setup screen or from the Protocol Spreadsheet. (You may, however, control trigger operation manually by selecting **Keyboard** trigger conditions; see Section 25.)

Select YES to display a data entry field and a rotating window for **BCC** (see Figure 26-4). When the trigger is true, the INTERVIEW will transmit any message you enter here (up to 37 characters), ending with the block–check selection you make in the **BCC** field. ASCII characters, hexadecimal entries, and control characters are valid in this field.

Any global constant declared on the Protocol Spreadsheet may be referenced as part of the **Xmit** string on a Trigger Setup screen. Enter the name exactly as it was declared on the Protocol Spreadsheet, and enclose it in double parentheses—for example, ((drop)). The double parentheses are special characters created by pressing CTRL–9 for ((, and CTRL–0 for)) .

Since the standard fox message, containing the set of upper–case alpha characters and the ten numerals, is predefined, you may reference it as a constant on a Trigger Setup screen. It need not be declared on the Protocol Spreadsheet. Enter it as follows: ((FOX)) .

Figure 26-4 Transmitted messages may be terminated with good or bad BCC's or an Abort.

(A) BCC

You have the option of following each text block transmitted from the Trigger Setup screen with a block–check calculation. Block checks are calculated according to your selections on the BCC Setup screen (see Section 10).

On the INTERVIEW screen, the final byte of the calculation appears as a highlighted overlay (G, B, or A), as long as you have selected **Rcv Blk Chk:** ON on the Line Setup screen (see Section 5). This selection is only available for synchronous and asynchronous formats. When **Rcv Blk Chk** is OFF for these two formats, block–check characters appear as they are actually transmitted.

The block–check symbol will be displayed for Bit–Oriented Protocols.

In the rotating **BCC** window, you may select GOOD, BAD, NONE, or ABORT. The default selection is GOOD.

1. *Good Block Check*. Select GOOD to terminate your text blocks with a correct block check. (Remember that not all transmissions are text blocks: a bisync poll will not receive a block check even if GOOD or BAD is selected.)

2. *Bad Block Check*. Select BAD to end your transmission with an erroneous block check. For Bit–Oriented Protocols, the bad BCC is CRC–16 instead of CCITT; for other formats, the bad BCC is an inverted good BCC.

3. *None*. When NONE is selected, no block check is sent at the end of the transmission. (For BOP transmissions, NONE has the same effect as ABORT.)

   You may cause messages to be sent in succession by different triggers, with no intervening block checks if you wish; however, at least one full character of idle (or $^{7}\varepsilon$ flag, in the case of Bit–Oriented Protocols) will be transmitted between blocks. When leads are switched (as indicated on the Interface Setup screen; see Section 12), the interface leads will be controlled between blocks.

4. *Abort*. This selection causes the message to which it is appended to abort before completion. When selected with Bit–Oriented Protocols, this action causes the INTERVIEW to transmit seven consecutive 1's at the end of the message. (For non–BOP transmissions, selecting ABORT has the same effect as selecting NONE.)

26.3 Internal Flags

Internal flags are bits that can be set on or off and sensed by triggers. Eight internal flag bits are shared among the Trigger Setup screens. Any combination of flag bits can be controlled by any trigger or combination of triggers.

> **NOTE:** The flag bits on the Trigger Setup can be controlled and monitored on the Protocol Spreadsheet, where they are referred to as **trig_flag** (see Section 31).

By default the **Flags** option is NO. Three other selections are available in the rotating window: SET, INC, and DEC.

(A) Set

When you select SET, a flag mask appears (see Figure 26-5). Use the arrow keys (→ and ←) to move the cursor to the bits you wish to set.

Enter a 1 or a 0 in any position you wish to set. Enter an X for "Don't Care." The trigger will not change the existing value of this bit.

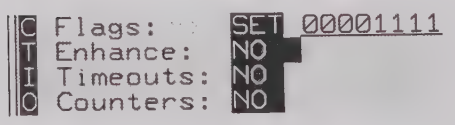


Figure 26-5 A set of eight flag bits may be set on any of the trigger menus.

(B) Increment

The internal flags, consisting of Flags 0 through 7, can be thought of as a binary number. This action increases the value of the flags by one each time the trigger is true. (Other trigger actions may change the value of the flag bits in the intervening period.) Incrementing flags is one technique for controlling recursive routines.

As the flag bits increment past 255, they roll over to zero.

(C) Decrement

This action decreases the value of the flag byte by one each time that the trigger is true. In the event that the flag decrements below zero, the value of the byte wraps to 255.

> **NOTE:** The value of the flag bits is continuously reset to zero when you enter Run mode.

26.4 Enhancing the Display

Triggers can be used to enhance display data selectively. Data on either or both sides of the line can be enhanced. Enhanced data is also stored in the character buffer with the enhancements for later review.

(A) BOTH, DTE, or DCE

Select **Enhance:** BOTH , DTE , or DCE to enable enhancement options (see Figure 26-6). BOTH indicates that enhancements will be turned on or off on both TD and RD data at the same time.

DTE pinpoints TD data for enhancement; DCE specifies RD data for enhancement.

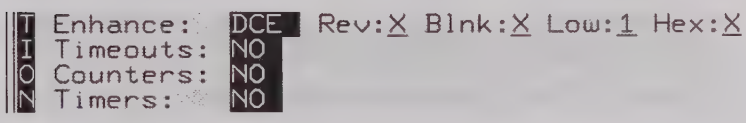


Figure 26-6 Data may be enhanced with respect to DTE, DCE, or both.

Four options, **Rev**, **Blnk**, **Low**, and **Hex**, appear to the right. To turn on an enhancement, enter a 1 on the line immediately following it. To turn off an enhancement, enter a 0 on the same line. When an X follows the enhancement, the trigger takes no action.

1. *Reverse image*. Reverse–imaged (**Rev**) characters are presented as dark letters on a lighter background.

2. *Blink*. **Blnk** causes data to blink on and off rapidly. This is the most conspicuous highlight for small portions of data.

3. *Low intensity*. **Low** has no effect on the plasma display. However, if you have installed a black and white monitor, it provides a low–intensity highlight for selected data.

4. *Hexadecimal*. When **Hex** is turned on, all data affected by the trigger is displayed in hexadecimal. Once data is stored in the buffer as hexadecimal, it remains in hexadecimal form.

(B) Color Enhancement

Color enhancement is controlled by the settings of three trigger enhancements: Reverse, Blink and Low. The three combined settings are mapped to color enhancements on the Miscellaneous Utilities screen as described in Section 18.

26.5 Controlling Timeouts

Each Trigger Setup screen can restart or stop either or both of the two timeout timers. These timers decrement from a value set on any of the triggers and, like flags and counters, serve as useful trigger conditions for internal program control.

When **Timeout:** YES is selected, identical new fields appear for Timeout #1 and Timeout #2. Both fields may be filled in on the same trigger.

(A) RESTART

Select RESTART to start or reset the timeout timer. The amount of time remaining on the timeout timer is entered in the data entry field provided (see Figure 26-7).

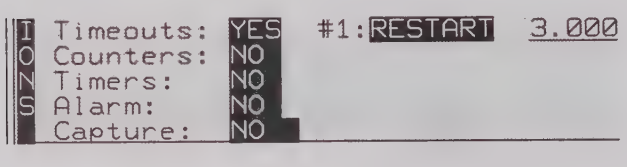


Figure 26-7 Timeout #1 activated to expire in three seconds.

1. *Entering timeout values*. The duration of the timeout is entered in seconds in the 5–character data–entry field provided. To enter a timeout value that is less than one second, use a leading zero before the decimal point, as follows: 0.25. The smallest valid timeout is 1 millisecond (0.001). The largest valid timeout is 65.535 seconds.

   Create a ten–minute timeout as follows: Start a timeout with a value of 60 seconds. When it expires, restart a similar timeout and increment a counter. When the counter equals ten, ten minutes will have elapsed.

(B) STOP

Select STOP to halt and clear the timeout timer, without causing the timeout to occur. If **Timeout** is selected as a trigger condition, the condition will not become true in this instance.

> **NOTE:** Timeouts created on a Trigger Setup screen can be monitored and controlled from the Protocol Spreadsheet. These timeouts are entered as **trig_timeout_1** and **trig_timeout_2** when referred to on the Protocol Spreadsheet.

26.6 Counters

Each Trigger Setup screen can control two counters. These counters can be unique to the trigger (controlled only by it), or they may be shared with other triggers, which can monitor them and change their values. As long as the same counter name is used, the same counter is invoked.

NOTE: Counter names used on the Protocol Spreadsheet also refer to these counters, if the names match any counter name on the Trigger Setup screens. This means that program control can be shared between these screens and the spreadsheet.

NOTE: Trigger Setup screens monitor counter values from 0 to 999,999. However, Protocol Spreadsheet triggers can monitor counter values up to 4,294,967,295.

(A) Menu Fields

When **Counters:** YES is selected, two sets of new menu fields, labeled 1st and 2nd, appear (see Figure 26-8).

1. *Counter name.* Enter the counter name in the field provided. The name may be up to eight characters long and must start with a letter. Upper– and lower–case alpha characters, numerals, and underscore (_) are legal in the other positions.

   When the name field is empty, the trigger takes no action for that counter field.

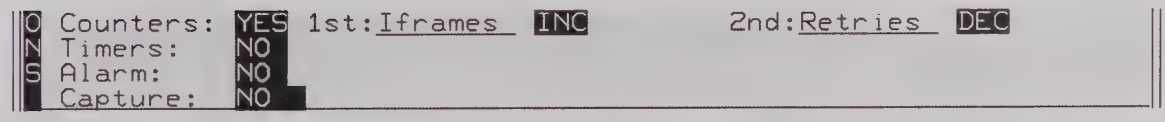


Figure 26-8 One counter is incremented, another decremented in this action.

2. *No.* The default selection is NO. It allows you to disregard one or both counters.

3. *Increment.* When INC is selected, each trigger occurrence adds 1 to the counter.

4. *Decrement.* When DEC is selected, each trigger occurrence subtracts 1 from the counter. When a counter decrements below zero, it wraps not to 9,999,999, but to the decimal equivalent of $2^{32} - 1$, the actual maximum value of a 32–bit counter. The seven least–significant decimal digits that appear on the Tabular Statistics screen are 4967295. The complete number is over 4 billion.

5. *Set.* Select SET in order to specify the value which the counter will take when the trigger becomes true. Then, enter the decimal value of the counter in the field provided. The field is six positions long, making it possible to set counters to a value from 0 to 999999. Any leading positions not specified in your entry will

be set to zero. This action does not cause statistical samples to be taken, nor does it reset last value, minimum value, maximum value, or average value for the counter. (Compare to Sample and Clear.)

6. *Sample*. This action causes the counter to reset to zero and causes measurements to be taken for last value, minimum value, maximum value, and average value. Refer to Section 21 for an explanation of how statistics are gathered and tabulated.

7. *Clear*. This action resets the counter to zero and also resets minimum value, maximum value, and average value for the counter.

26.7 Timers

Two timers are shared among the Trigger Setup screens. While these timers are not available as trigger conditions, they can be run and sampled as trigger actions. When timers are invoked by triggers, their values can be tracked on the statistics screens (see Sections 21 and 22).

> **NOTE:** Timer names referred to on the Protocol Spreadsheet may also be used on Trigger Setup screens. Thus, timer control of programs is shared between these screens and the spreadsheet.

(A) Menu Fields

The default timer selection is NO. When YES is selected, two identical subfields appear, for Timer 1 and Timer 2 (see Figure 26-9).

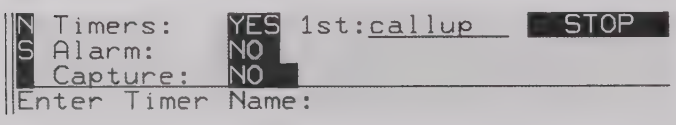


Figure 26-9 One or two timers may be controlled by the same trigger (second field not shown).

1. *No*. The default selection for each Timer is also NO. This allows trigger action to disregard both timers or to focus on one timer, if necessary.

2. *Restart*. When selected, RESTART causes the timer to reset to zero and begin incrementing. RESTART does not cause statistical measurements to be taken. (Compare to SAMPLE and CLEAR.)

3. *Stop.* The STOP action suspends the timer and allows it to retain its value. The timer may be started again at this value by a CONTINUE action on another trigger.

4. *Continue.* CONTINUE, when selected, causes the specified timer to increment, starting from the value at which it was stopped.

5. *Sample.* The SAMPLE action resets and stops the timer. Prior to resetting the timer, its value is read as a "last" value and passed along for other statistical measurements. Refer to Section 21 for an explanation of how statistics are gathered and tabulated.

6. *Clear.* The CLEAR action resets the current value, the last value, the minimum value, the maximum value, and the average value of the timer. Refer to Section 21 for an explanation of statistical measurements.

26.8 Alarm

The alarm is a short beep. The alarm is useful for calling your attention to the data being analyzed, especially when the situation of interest occurs infrequently. When you select **Alarm:** YES, it is sounded each time the trigger becomes true.

26.9 Capture of Data in the Screen Buffer

Capture of character–oriented data to the screen buffer can be stopped and restarted by triggers, using the **Capture** action (see Figure 26-10). When capture is turned off, data is neither presented to the screen nor stored in the buffer.

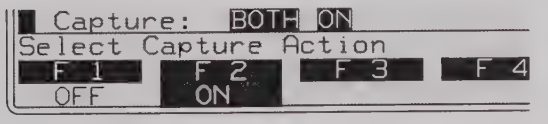


Figure 26-10 Data capture to the screen buffer can be controlled by triggers.

(A) NO, BOTH, DTE, or DCE

The default **Capture** selection is NO. This represents no change; that is, the trigger does not influence character buffer capture. By default, data is continuously captured in the character buffer.

Select BOTH to control capture to the character buffer for TD and RD data at the same time. Select DTE to control only TD data; DCE to control only RD data.

1. *OFF, ON.* Select OFF to suppress data from the screen buffer. Select ON when another trigger has turned off capture and you wish to begin storing data in the buffer again.

27 The Trigger Summary Screen

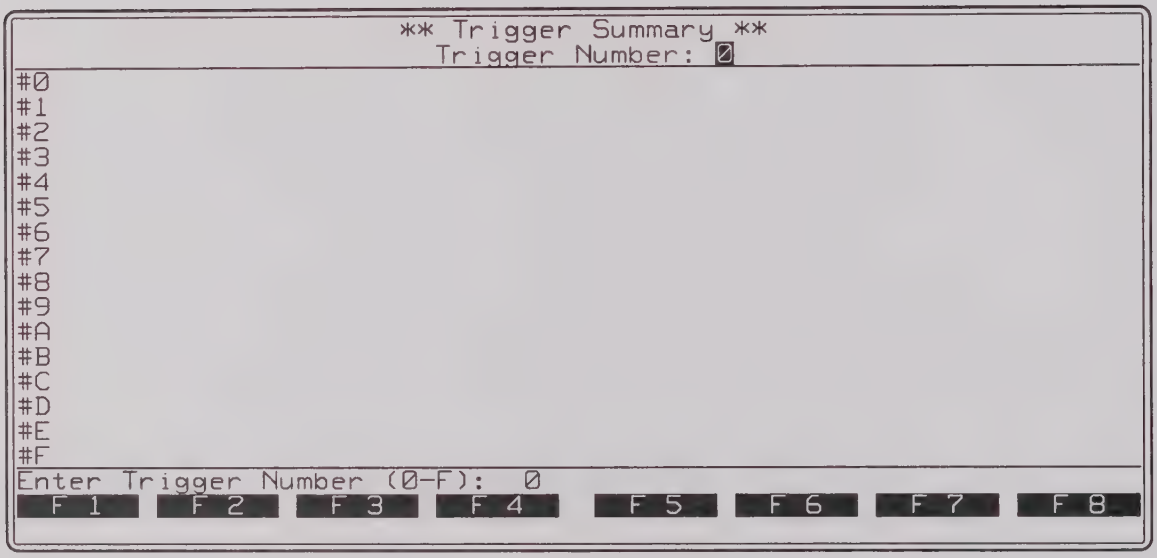


Figure 27-1 Default Trigger Summary screen.

27 The Trigger Summary Screen

The Trigger Summary screen is the access screen to all Trigger Setup screens. The default Trigger Summary screen is shown in Figure 27-1. Call up the Trigger Summary screen by pressing the function key marked TRIGS on the main Program Menu. With the summary displayed, access any trigger by typing the number of the desired screen (0 through F). To see a synopsis of configured Trigger Setups, you may return to the summary screen from any trigger menu by pressing PREV PAGE.

Entries you make on any of the 16 Trigger Setup screens appear on the summary in abbreviated form. Each setup screen is allotted a one–line summary. A summary of conditions appears on the left–hand side of the line; a summary of actions appears on the right–hand side of the line. The summary for Trigger Setup screen 0 (Figure 27-2) is shown in Figure 27-3.

Abbreviations for possible Trigger Setup conditions are listed in Table 27-1. Abbreviations for Trigger Setup actions are given in Table 27-2.

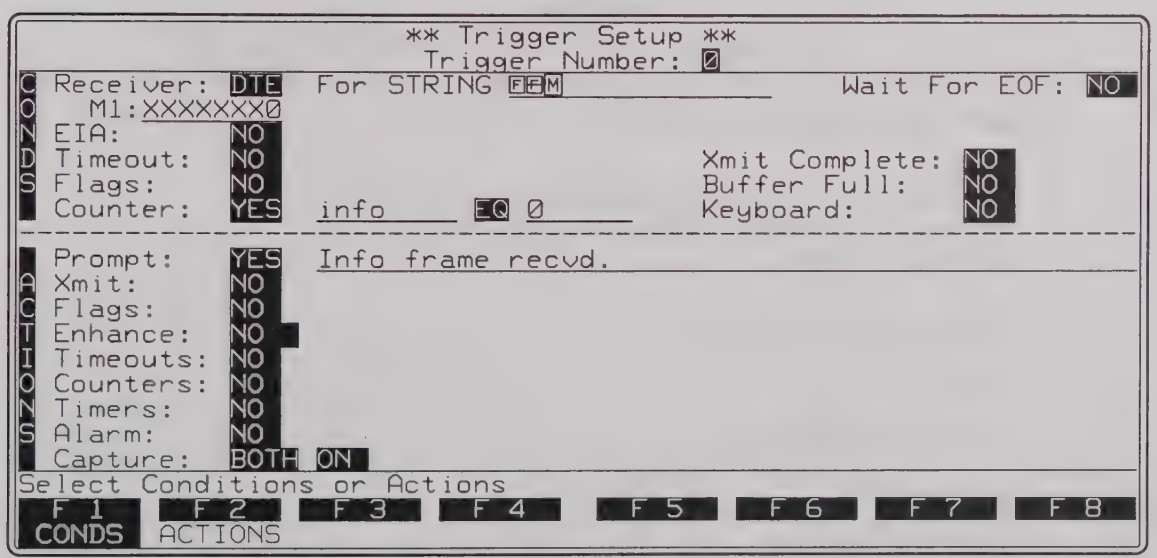


Figure 27-2 Entries on each Trigger Setup screen are indicated on the Trigger Summary.

NOTES:

Abbreviations displayed on the Trigger Summary screen are not necessarily keywords and should not be referred to when you are typing entries on the Protocol Spreadsheet.

When multiple conditions or actions are selected on a single Trigger Setup screen, the summary screen may not be able to show all selections; however, as many conditions and actions as possible will be displayed in the available space.

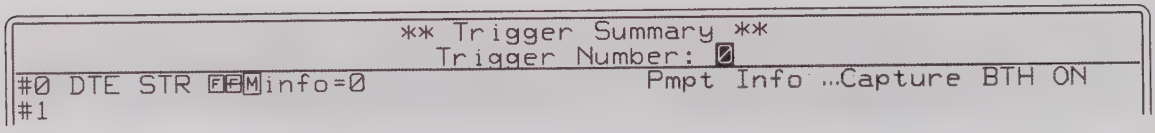


Figure 27-3 Summary of entries made on Trigger Setup screen 0, shown in previous figure.

Table 27-1
Abbreviations, Trigger Summary Conditions

Receiver (Word does not appear on summary.)

DTE, DCE
STR: String, **1OF:** One of (Character string also appears for **STR** and **1OF**.),
G: Good BCC, B: Bad BCC, **PrErr:** Parity Error, **FrErr:** Frame Error,
A: Abort, M: Bit Mask

EIA:
RTS, **CTS**, **CD**, **DTR**, **DSR**, **RI**, **UA**

TimeOut 1, 2

Xmit_Cmpl: Transmission Complete

Buffr_Ful: Buffer full

Flag: (Value only appears.)

Counter: (Name and value only appear.)

KeyBd: Keyboard (Key indicated.)

Table 27-2
Abbreviations, Trigger Summary Actions

Pmpt: Prompt (Prompt string also appears.)

Xmit: Transmit (Xmit string also appears.)
G: Good BCC, B: Bad BCC, A: Abort
(Nothing appears if no BCC is selected.)

Flag: INC: Increment, **DEC**: Decrement (Value only appears if selection is **SET**.)

ENH: Enhance Display
BTH: Both DTE and DCE, **DTE**, **DCE**
REV=: Reverse, **BLN=:** Blink, **LOW=:** Low, **HEX=:** Hexadecimal

TO #1, **TO #2:** Timeout #1 or 2
RST: Restart, **STP:** Stop

*Counter: (*Only name and value appear.)
INC: Increment, **DEC:** Decrement, **=:** Set
SMP: Sample, **CLR:** Clear

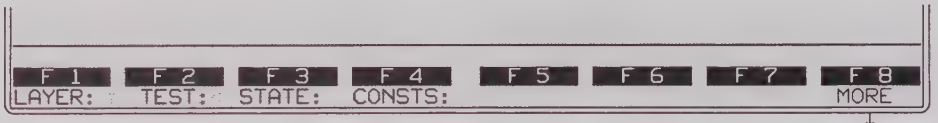
TM: Timer (Name also appears.)
RST: Restart, **STP:** Stop, **CNT:** Continue, **SMP:** Sample, **CLR:** Clear

Alarm: Audible Alarm

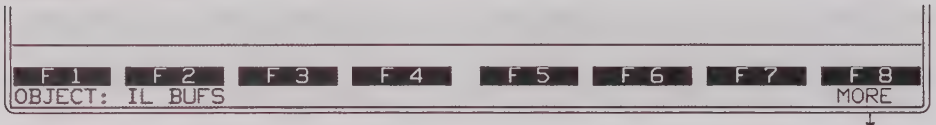
Capture: Capture Memory
BTH: Both DTE and DCE, **DTE**, **DCE**
ON, OFF

28 Programming Blocks

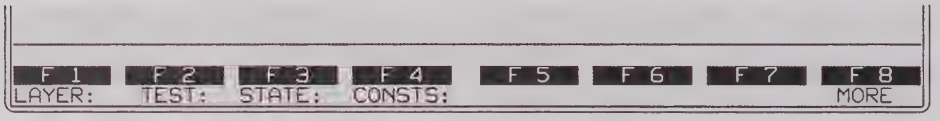
1. These keys are at highest level:



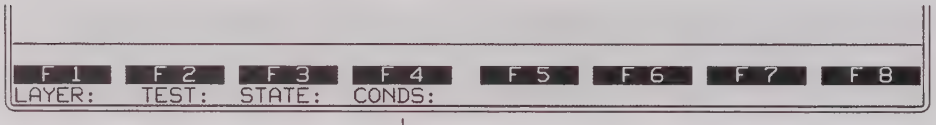
2. Press MORE to obtain:



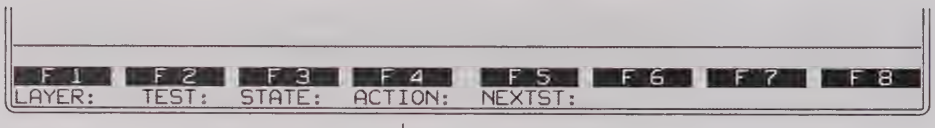
Press MORE to return to:



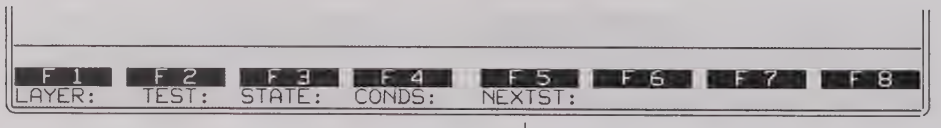
3. Press STATE: enter name and press RETURN or the spacebar to obtain:



4. Press CONDS: (Conditions) to obtain variable set of Conditions.* Press F8 to access alternate racks of Conditions keys. Complete entries, then press DONE to obtain:



5. Press ACTION: to obtain variable set of Actions.* Press F8 to access alternate racks of Actions keys. Complete entries, then press DONE to obtain:



Press NEXTST:, enter name of next state, and press RETURN or the spacebar *to obtain rack 3.*
(Pressing CONDS: obtains variable sets of conditions.)

*Conditions and Actions available depend on what protocols are loaded and what layer number you have specified. The following selections are always available:

GENERAL CONDITIONS

ENTER_STATE	TIME
TIMEOUT	FLAG
KEYBOARD	ON_SIGNAL
BUFFER_FULL	
COUNTER	

GENERAL ACTIONS

FLAG	LOAD_PROGRAM
ACCUMULATE	PROMPT
SIGNAL	PRINT
COUNTER	TRACE
TIMER	ALARM
TIMEOUT	RECORD

Figure 28-1 Function key hierarchy, Protocol Spreadsheet.

28 Programming Blocks

The Protocol Spreadsheet is a highly flexible programming approach which enhances trigger conditions and actions provided on the Trigger Setup menus, furnishes new general options, and incorporates protocol–specific conditions and actions on a layer–by–layer basis.

28.1 Before You Begin a Spreadsheet Program

Be certain prior to programming that you have loaded the Personality Packages for the protocols you will be testing. Automatic protocol options are part of each layer's Personality Package. These packages are loaded from the Layer Setup screen as described in Section 8.

Check the configuration of the various Test Setup screens before you test or save your program, since the behavior of the INTERVIEW during testing is influenced by setup selections.

28.2 Creating a Spreadsheet Program

Press [F3] to access the Protocol Spreadsheet from the Program Menu. Any program which you have loaded from the File Maintenance screen appears on the spreadsheet. If no program has been loaded or created, the Protocol Spreadsheet, since it is a free–form menu, will be blank except for a header line, function key labels, and tildes (~) down the left side of the screen. Tildes always mark the end of your program file.

(A) Two Sets of Function Keys: Programming and Editing

Two full sets of softkeys are active with the spreadsheet. One set of softkeys groups available programming options, including keywords (LAYER:, TEST:, CONDITIONS: etc.). The alternate set groups sophisticated editing functions.

These editing functions, which complement the editing keypad, are accessible from the spreadsheet at any time. Press [EDIT] to activate edit softkeys. Press [EDIT] again to return to program softkeys. For a discussion of editing options, refer to Section 30.

(B) Programming Functions

Use the programming softkeys to make program entries, from the highest level of the program (OBJECT), down to individual trigger conditions and actions and their subfields. Softkeys guide you as you create your program by listing available options

and providing correct syntax wherever possible. (Errors are indicated by strikeover of incorrect text as you make your program entries.) For each level of function keys, a cue near the bottom of the screen explains selectable options or prompts you for keyboard entry.

Program softkeys are immediately available when you enter the spreadsheet. The hierarchy of the program softkeys is shown in Figure 28-1. The conditions and actions listed, which are always available, are explained in Section 31. Other trigger conditions and actions are added when protocol packages are loaded. Because protocols are layer–specific, trigger options will vary from layer to layer. For each LAYER block within your program, different options are likely to appear when you enter the keyword CONDITIONS or ACTIONS. For more information on the specific trigger options enabled by a protocol, consult the section devoted to that protocol (see Table of Contents, Section 36 and following.)

You also have the freedom of typing in any program entry, if you prefer, as long as you enter the block identifiers and conditions and actions keywords as they would be posted on the screen by softkeys. Syntax errors still are automatically highlighted by a strike–through.

> **NOTE:** Softkey labels are not necessarily legal spellings on the spreadsheet. Pressing the function key usually posts an expanded keyword on the screen. Use these expanded keywords when typing entries.

1. *Successive racks of softkeys.* The rack of softkey options at the bottom of the spreadsheet screen (or the instructional prompt on the third line up from the bottom, or both the option rack and the prompt) will change automatically each time you *complete* a keyword entry. Keyword entries are complete when you make them via softkey *or* when you type the keyword followed by a space or a hard RETURN. (Pressing the softkey has the same effect as typing the keyword and then typing a space to complete the entry.)

   Programming movement is generally *down* the tree of softkey racks, as in this series of keywords:

   ```
   CONDITIONS: EIA CTS ON
   ```

   Each of the four keywords was selected from a rack of options, and each succeeding rack is a step farther down the "branch." The rack that follows ON, however—listing RTS, CTS, CD, and other EIA leads—is back up the tree, since "there is nowhere to go but up," and since a trigger with multiple EIA conditions (like the following) is valid.

   ```
   CONDITIONS: EIA CTS ON CD OFF
   ```

2. *Additional racks of valid softkeys.* There may be many more keywords that are valid to enter at a given point in the program than are showing on one rack of softkeys. Additional racks may be accessible via the F8 softkey (MORE); and *higher* racks are generally available via the DONE key. In this series, DONE was pressed following the softkey for ON to access the softkey for COUNTER:

```
CONDITIONS: EIA CTS ON
  COUNTER xmit LT 6
```

   In the next series of keywords, DONE was pressed *twice* following the softkey for ON, to access the softkey for ACTIONS:

```
CONDITIONS: EIA CTS ON
ACTIONS: SEND " ((FOX)) " GOOD_BCC
```

   Note that DONE is not a valid keystroke following CTS above, since the condition syntax is not "done." Whenever it is not valid to move to a rack of softkeys higher up the tree, DONE produces an alarm tone.

   Note also that it is never necessary to press DONE if you are typing in your keywords directly from the keyboard. DONE merely changes the *rack* that is showing, not the entire set of keywords that is valid. A keyword does not have to be showing to be typed in legally.

3. *Insert mode versus overstrike mode.* Touch–typists in particular should be aware that the Protocol Spreadsheet has an insert mode as well as an overstrike mode. The insert mode is invoked by either of two keys, INSERT CHAR or INSERT LINE. When the mode is enabled, the word **<Insert>** appears at the top left of the screen. In insert mode, the programmer types in a block of data while succeeding text is pushed forward with every keystroke.

   Press INSERT CHAR (but not INSERT LINE) a second time to exit *insert* mode and return to *overstrike* mode.

The remainder of this section is devoted to the fundamentals of program structure and to programming components available on the Protocol Spreadsheet which are independent of trigger options.

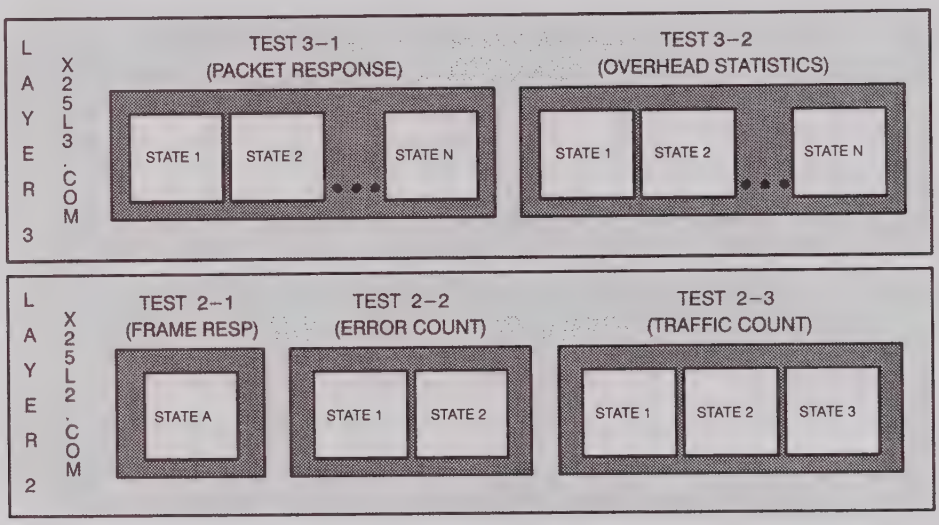


Figure 28-2 Discrete states inhabit separate tests at separate layers.

28.3 Program Structure

The components of the INTERVIEW's programming model, introduced in Section 24, are integrated into a spreadsheet program as discrete blocks according to specific structural rules. Compare the abstract program model in Figure 28-2 to the spreadsheet program outlined in Figure 28-3.

(A) Block Identifiers

The INTERVIEW's compiler must respect the distinction between one layer and the next and between one test and the next. Further, it must group triggers into designated states and track the transition from one active state to another. To indicate the boundaries of these various blocks, specific keywords are used. Each block normally begins with an identifier in upper−case letters, (optionally) followed by a colon. A block ends when a new block identifier is inserted in the program.

> **NOTE:** The identifier must not be enclosed in quotes (that is, must not be part of a text string) if it is intended as a block delimiter.

Available program blocks, from largest to smallest, are described in subsequent paragraphs. The valid block identifier for each is printed above its description.

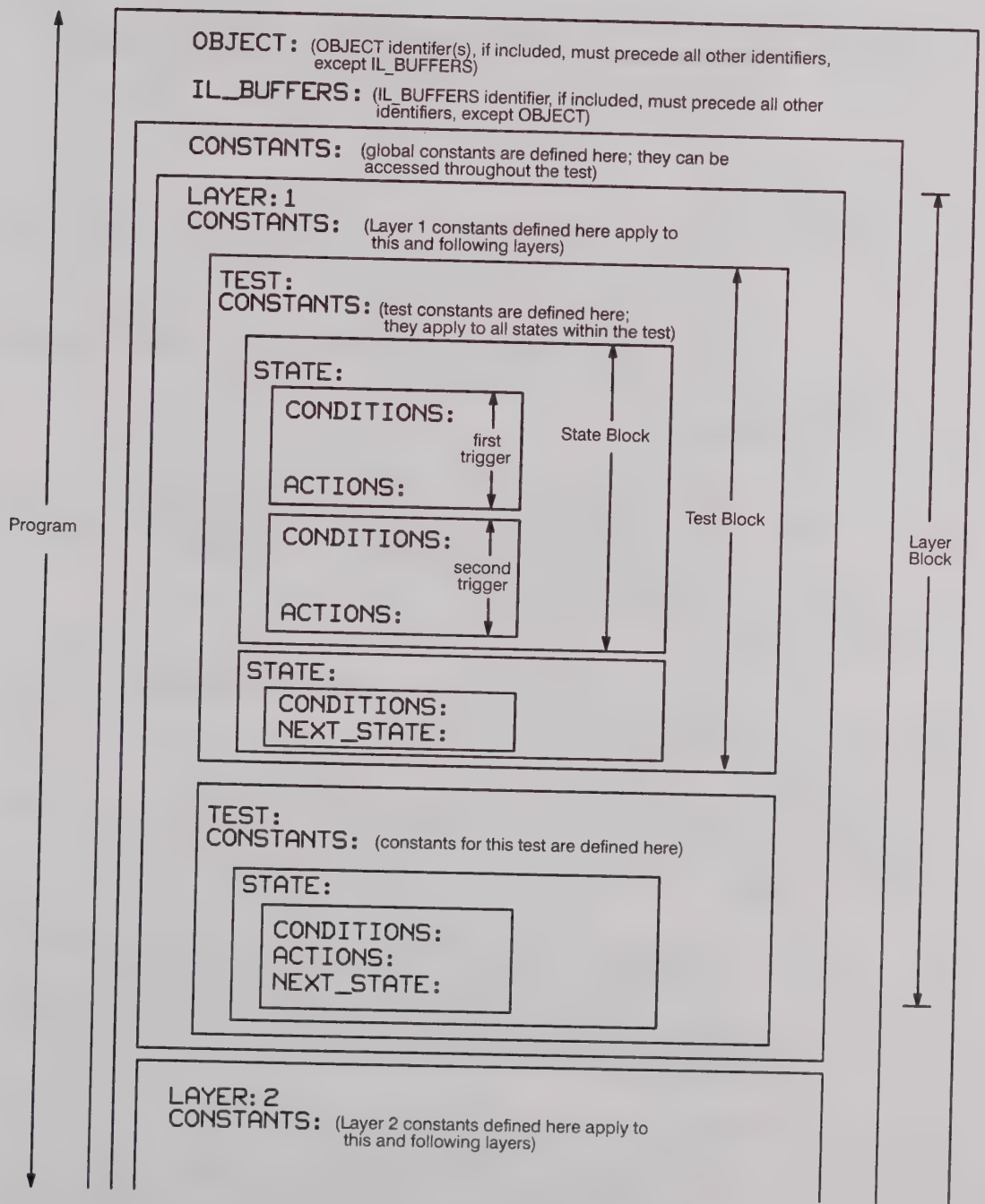


Figure 28-3 Program Structure. Component blocks begin with a keyword.

OBJECT:

1. *Referencing linkable–object files.* Use the OBJECT block–identifier to access the compiled code in a linkable–object file. See Section 28.4 below. The OBJECT identifier(s) must appear at the top of the Protocol Spreadsheet. IL_BUFFERS is the only identifier which may precede OBJECT. C regions or spreadsheet comments may also precede the OBJECT block identifier.

IL_BUFFERS

2. *Configuring the number/size of IL buffers.* Interlayer (IL) message buffers are used to pass data up the layers as it is received and down the layers as it is transmitted. Press the IL BUFS softkey to set the number and size of the IL buffers. The IL_BUFFERS identifier(s) must appear at the top of the Protocol Spreadsheet. OBJECT is the only identifier which may precede IL BUFS. C regions or spreadsheet comments may also precede the IL_BUFFERS block identifier.

CONSTANTS:

3. *Defining constants.* There are three legal locations for the definition of a constant: in the opening lines of a program, at the beginning of a layer, or at the beginning of a test. The relative placement of a constant's definition within a program determines its scope, or active range. For a complete discussion on constants, refer to Section 29.

LAYER:

4. *Layers.* The largest block, the layer, corresponds to the OSI model. There may be up to seven layers in any test.

TEST:

5. *Tests.* A layer may contain any number of simultaneous tests. Every test resides inside a layer.

STATE:

6. *States.* In turn, each test may contain any number of states. A state always resides inside a test. Only one state in each test is active at one time.

   Within each state, there may be a number of triggers. A trigger always resides inside a state. Each trigger is composed of a conditions portion and an actions portion.

CONDITIONS:

7. *Trigger conditions.* A single condition or a group of conditions is normally listed after the CONDITIONS identifier. Rules for grouping trigger conditions, as well as the meaning of each trigger condition, are explained in Sections 31 and 32.

ACTIONS:

8. *Trigger actions*. The ACTIONS identifier precedes the list of trigger actions. This list may be empty, or it may include one or several trigger actions. The various trigger actions are described in Sections 31 and 32.

NEXT_STATE:

9. *Next state*. The identifier NEXT_STATE, explained in the following paragraphs, can replace the ACTIONS: identifier in a trigger if there are no other actions; or it can follow the ACTIONS: identifier to indicate that branching to another state is one of several actions taken by the trigger.

(B) Run–time Transitions Between States

Run–time transitions between states are controlled by triggers. To indicate a run–time branch from one state to another, use the NEXT_STATE action, followed by (a) the name of the state you wish to go to, or (b) the NEXT token, indicating whatever state happens to follow sequentially in the spreadsheet program.

You may use a NEXT_STATE action once per trigger and as many times as needed in one state to allow for multiple branching possibilities.

When two triggers come true at the same time and both potentially result in branching to another state, the trigger which is checked last (the last trigger sequentially displayed on the spreadsheet) will cause branching to the state it names. (The first trigger will not cause branching.)

Look at the two triggers shown in the example which follows. The first searches for any SDLC Information frame. The second searches for an Info frame with a particular frame address. By definition, whenever the second trigger is true, the first trigger is also true. When an Info frame with the correct address is received, the second trigger causes the test to branch to the State **respfrm**. However, if these triggers are reversed as shown in the second example, the test *always* branches to the State **otherfrm**, regardless of the frame address.

correct order

```
STATE: frmadd
    CONDITIONS: DTE INFO
    NEXT_STATE: otherfrm
    CONDITIONS: DTE INFO ADR=C1
    NEXT_STATE: respfrm
```

wrong order

```
STATE: frmadd
    CONDITIONS: DTE INFO ADR=C1
    NEXT_STATE: respfrm
    CONDITIONS: DTE INFO
    NEXT_STATE: otherfrm
```

(C) Recommended Format

The format of a Protocol Spreadsheet is entirely flexible. The only rule is that block identifiers must (with rare exception) be included in the program to designate boundaries between programming blocks.

The following is a suggested program format. To create a visual distinction, the keywords which define program blocks are placed at the beginning of a line. Smaller blocks are indented to show that they reside within a larger block. An automatic indent feature, described in Section 30, is included as an editing function and is turned on by default.

```
LAYER: 1
   TEST: echo_msg
      STATE: message
         CONDITIONS: DTE STRING "hello"
         ACTIONS: PROMPT: "Spreadsheet trigger true."
         NEXT_STATE: echo

      STATE: echo
         CONDITIONS: DCE STRING "hello"
         ACTIONS: PROMPT " Echoed message received"
         NEXT_STATE: message
```

(D) Omitted Block Identifiers

It is recommended that, for ease of tracking a program, block identifiers be placed at the beginning of every block. However, in brief programs, certain block identifiers may be omitted.

It is, in fact, possible for a program to begin with a STATE identifier. The compiler then assumes that you have begun the first test inside the first layer of the program. To start another program block, you must use a STATE, TEST, or LAYER identifier.

> **NOTE:** Any constant declared in the opening lines of a test which omits the LAYER and/or the TEST keyword is still a global constant, as long as it precedes a STATE or CONDITIONS identifier.

28.4 Compiled Spreadsheet

Using the Compile command on the File Maintenance screen, you can compile and save the contents of the Protocol Spreadsheet in a linkable–object file. Later, this program can be combined with an active spreadsheet program. To do so, simply reference the file at the top of the Protocol Spreadsheet.

(A) The OBJECT Block–Identifier

Use the OBJECT block–identifier on the Protocol Spreadsheet to access the compiled spreadsheet code in a linkable–object file.

> **Note to C Programmers:** The OBJECT identifier may also be used to access definitions for user routines. Refer to Section 55.4(C).

1. *Placement.* The OBJECT block–identifier(s) must appear at the top of your spreadsheet program, ahead of any other identifier (except IL BUFFERS). Access the OBJECT: softkey by pressing MORE on the initial rack of softkeys. Notice that the MORE, IL BUFS, and OBJECT: softkey tokens are not available once any other programming block–identifier has been selected.

   NOTE: Use OBJECT in your active spreadsheet program only. Do not incorporate it in a spreadsheet that will be compiled and saved as an LOBJ file. Although the code will compile, the referenced LOBJ file will not be read.

2. *Format.* The format for the OBJECT block–identifier is as follows:

   ```
   OBJECT: "filename.o"
   ```

   The identifier references only one linkable–object file, but you may include as many OBJECT identifiers as you wish.

   The relative or absolute pathname of the linkable–object file is enclosed in quotation marks.

3. *Search rules for linkable–object files.* As your spreadsheet program compiles, the INTERVIEW's filing system is searched for the linkable–object files referenced in OBJECT identifiers.

   - If the referenced LOBJ filename begins with *FD1/*, *FD2/*, or *HRD/*, the INTERVIEW interprets it as the absolute pathname and makes only that one search.
   - Pathnames beginning with a / indicate that the root directory on each drive should be the beginning point of the search. The drives are searched in the following order: current drive, FD1, FD2, and HRD.
   - Otherwise, the name may be a one–word filename or a relative pathname which includes the directories leading to the file. The highest directory in a relative pathname must reside in the current directory or in one of the */lib* subdirectories. The following directories—and only the following directories—are searched, in the order given:

1. current directory on the current drive (indicated on the File Maintenance screen)
2. */usr/lib* on the current drive
3. */sys/lib* on the current drive
4. *FD1/usr/lib*
5. *FD2/usr/lib*
6. *HRD/usr/lib*
7. *FD1/sys/lib*
8. *FD2/sys/lib*
9. *HRD/sys/lib*

If the pathname is not located in any of these directories, the program will not compile and an error message will be returned to the operator.

(B) Compiled LOBJ Code is Combined with Spreadsheet

During compilation, the compiled spreadsheet in the LOBJ file is combined with your active spreadsheet program. This means that the LOBJ code must be compatible with the current menu setups and spreadsheet program—as though the source code of the LOBJ file were actually present in the spreadsheet buffer.

(C) Counter and Flag Conditions

Special consideration is given to COUNTER and FLAG conditions during the Compile SPREADSHEET operation. The system identifies the condition as either transitional or status. (See Section 31.2.) If it is used both ways in the same spreadsheet file, it will always be identified as transitional.

Within a single spreadsheet program, you may reference more than one LOBJ file which uses the same COUNTER or FLAG. If one of the files uses the COUNTER (or FLAG) as a transitional condition, however, all other referenced files containing the same COUNTER (or FLAG) must also use it as a transitional condition at least once. This rule ensures that each action on the specified COUNTER (or FLAG) will consistently trigger the appropriate COUNTER (or FLAG) conditions.

(D) Advantages of Compiled Spreadsheet

Linkable–object files assist the programmer in efficiently using the INTERVIEW's memory and spreadsheet buffer.

- When commonly utilized conditions and actions are saved in linkable–object files, space in the spreadsheet buffer otherwise dedicated to this purpose can be used for additional programming.
- Since the code in LOBJ files has already been compiled, the INTERVIEW can support a larger program without a corresponding increase in compilation time.
- The spreadsheet code in a linkable–object file is transparent to the configuration of the unit. LOBJ files created on one unit can be used on a unit configured differently, as long as the code is compatible with the various menu parameters.

28.5 Configuring the Size/Number of IL Buffers

Interlayer (IL) message buffers are used to pass data up the layers as it is received and down the layers as it is transmitted. (See Section 24 on the layered–programming model and Sections 34 and 62 for more information on the uses of IL buffers.) The INTERVIEW allocates IL buffers, as needed, to pass data between layers. Then, the buffers are automatically erased and used again. In this way, the INTERVIEW maximizes its use of available memory space. Without these reusable buffers, data in Run mode would quickly eat up all of the memory in the unit. (See Section 62.3(A) for information on manipulating IL buffers.)

IL buffers contain the data itself or point to the memory location (outside the buffer) of the data. It follows, therefore, that the larger the IL buffer, the more data it can hold. By default there are 16 IL buffers that can be in use at a given time. The size of each buffer is 4,096 bytes.

When you are performing emulation with windowing, you can quickly use up these sixteen buffers. Once all buffers are in use, additional data is lost. To prevent this from happening, you may want to reconfigure the number and size of IL buffers.

Press the IL BUFS softkey to set the number and size of the IL buffers. Figure 28-4 shows the softkey selections. Select one of seven number/size combinations for the INTERVIEW's IL buffers. The default selection is 16/4K. This means that the INTERVIEW will have a maximum of 16 IL buffers in use at a given time, each one 4,096 bytes (4 Kbytes). This size, and all others, includes a 32–byte buffer header.

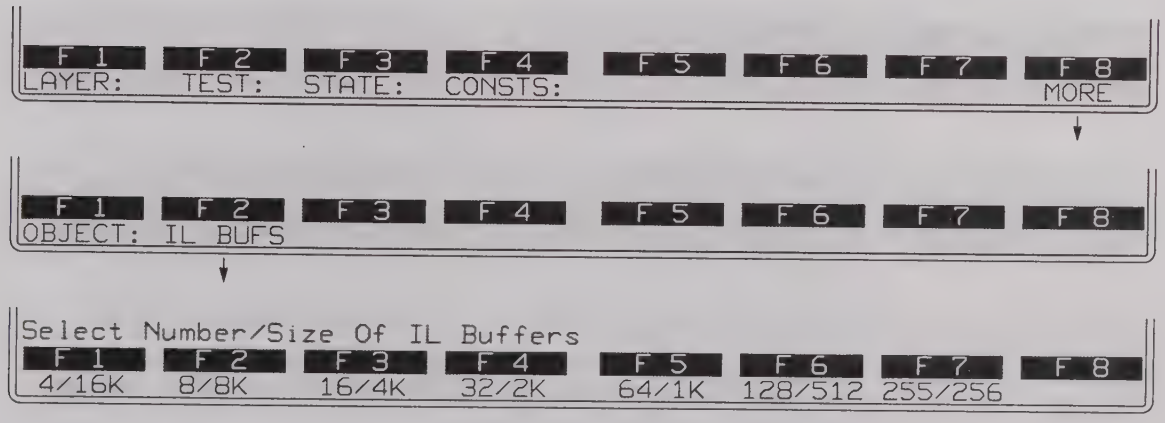


Figure 28-4 Softkey path to the seven number/size combinations for IL buffers.

Note to C Programmers: There are two preprocessor directives—*#pragma il_buffers* and *#pragma il_buffer_size*—which the C programmer may also use to configure the IL buffers. These directives provide additional flexibility. See Section 62.1(A).

Notice that each number/size combination utilizes 65,535 bytes (64 Kbytes) of RAM. This total represents the maximum amount of RAM that can be allocated for IL buffers from the Interlayer Buffers menu.

NOTE: Keep the following points about object–file compatibility in mind when setting the number/size of IL buffers:

- If the number of buffers is less than or equal to 16, the file will load and operate on any unit with equivalent hardware, with a software release earlier than 8.00.
- If the number of buffers is greater than 16, the file cannot be loaded on a unit with a software release earlier than 8.00.
- An object file generated under a software release earlier than 8.00 will run on software revision 8.00, or higher, with 16 buffers of 4 Kbytes each (the default).

28.6 Comments in a Spreadsheet Program

You may write comments to yourself or to others who may view your spreadsheet program. Comments begin with /* and end with */, as in the examples below. Use comments generously throughout spreadsheet programs. Since comments are ignored by the compiler, they do not affect the compilation time of the program.

(A) Characteristics

1. *Valid characters.* When an opening /* is detected by the compiler, everything that follows is disregarded until a closing */ is encountered. This means that all hexadecimal, control, and ASCII characters (or character combinations) are valid in comments. The F, S, X, and not–equal symbols are also legal entries.

   Two entries are not legal in comments. The first is the M symbol. It cannot be used because it is not a valid Protocol Spreadsheet entry. (Bit masks on the spreadsheet are delimited by « and »). An alarm will sound if you try to use the bit–mask symbol. The second invalid entry is the closing delimiter (*/). An embedded */ causes the comment to be ended prematurely. Since the remainder of the comment (and the programmer's *intended* closing */) is a syntax error, the program will not compile.

2. *Length.* For practical purposes, make comments as long as you wish. They may span several lines, or they may be empty.

3. *Location on spreadsheet.* Comments may be placed within any of the programming blocks: OBJECT, IL_BUFFERS, CONSTANTS, LAYER, TEST, STATE,

CONDITIONS, ACTIONS, or NEXT_STATE. In CONDITIONS blocks, however, they must appear with at least one valid condition. The following CONDITIONS block containing only a comment will cause compilation to be aborted:

```
STATE: message
   CONDITIONS: /* KEYBOARD " "*/
   ACTIONS: SEND "((FOX)) " GOOD_BCC
```

Since the compiler ignores anything inside the /* */ delimiters, it can find nothing in the CONDITIONS block. When you go to the Protocol Spreadsheet and search for error messages, the following message will be displayed: *"Empty Conditions Section."*

Comments may not be embedded within a keyword. This program also will not compile:

```
STATE: message
   CONDITIONS: KEY/* This comment will cause a syntax error*/BOARD " "
   ACTIONS: SEND "((FOX)) " GOOD_BCC
```

(B) Using Comments

Comments are particularly useful in describing the purpose of a programming block. Let's return to the two programming examples in which branching to another state occurs based on DTE Info−frame addresses. The following comment makes the programmer's intentions clear.

```
STATE: frmadd

/*  If a DTE INFO frame has an address of C1, go to state "respfrm."  For all other DTE INFO
frames, go to state "otherfrm." */

   CONDITIONS: DTE INFO
   NEXT_STATE: otherfrm
   CONDITIONS: DTE INFO ADR= C1
   NEXT_STATE: respfrm
```

Comments can be useful debugging tools. Suppose the same comment appeared in the programming example with the order of the two triggers reversed.

```
STATE: frmadd

/*  If a DTE INFO frame has an address of C1, go to state "respfrm."  For all other DTE INFO
frames, go to state "otherfrm." */

   CONDITIONS: DTE INFO ADR= C1
   NEXT_STATE: respfrm
   CONDITIONS: DTE INFO
   NEXT_STATE: otherfrm
```

With the comment present, it is easier to identify the discrepancy between the programmer's expectations and the actual program.

29 Constants

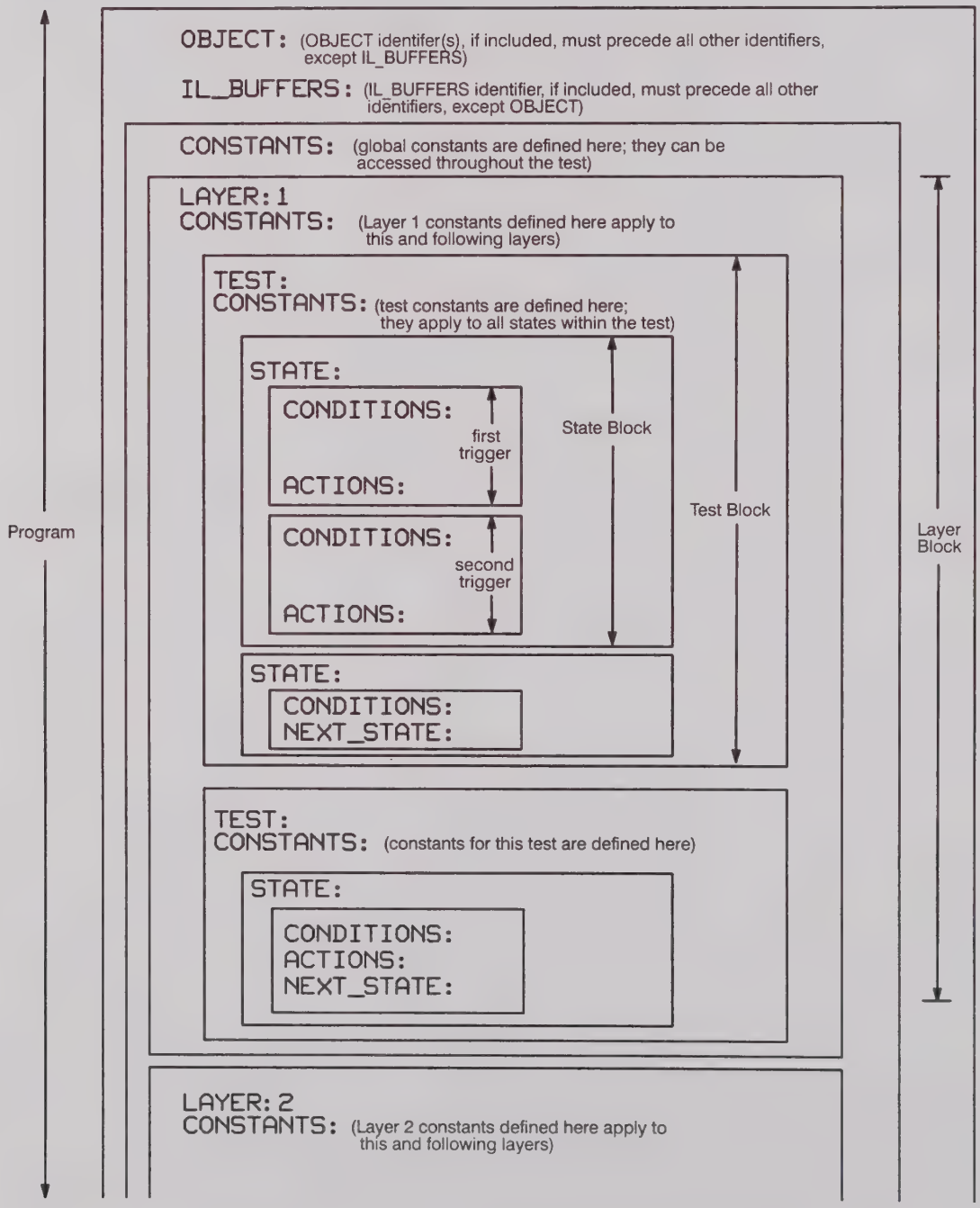


Figure 29-1 Constants may be defined in three different locations: before the first layer, after a layer identifier, or after a test identifier.

29 Constants

The Protocol Spreadsheet permits the use of constants as a means of simplifying the creation and modification of test programs. A constant is merely a symbolic reference to a predefined string of characters.

29.1 Definition of Constants

There are three legal locations for the definition of a constant: in the opening lines of a program, at the beginning of a layer, or at the beginning of a test (see Figure 29-1). Constants must always be defined at the beginning of the programming block in which they are referenced. A test−level constant may not be preceded by a lower−level block identifier (STATE, CONDITIONS, or ACTIONS). A constant definition or definition block must be followed by another keyword.

A constant definition begins with the identifier CONSTANTS. A colon (:) may follow. The constant name is then entered. Next comes the definition, a text string which the constant represents. The constant name and text string may be separated by an equal sign (=). The text string is enclosed in double quotes. Each constant definition comprises a single logical line. Logical lines wrap as needed to subsequent lines on the spreadsheet, but they do not contain hard returns. See Section 30.1(A).

More than one constant can be defined following a single CONSTANTS identifier. Following is the suggested format for a constant definition block. A constant definition block (whether it contains one or more definitions) must be followed by another block identifier.

```
CONSTANTS:

   cmd_adr  = "03"
   resp_adr = "01"
   resend   = "1.5"
   retries  = "10"
```

To include quotation marks or backslashes in the definition string of a constant, precede each with a backslash escape−character (\). Here, for example, is the constant definition of a general poll:

```
CONSTANTS: drop_A= '' AA\'' \'' E0 ''
```

The backslash will be deleted by the parser when the constant definition is scanned:

```
AA'' '' E0
```

If the constant is contained in a search or transmit string, it will not be scanned for the escape character or for closing quotation marks. The characters shown above represent the expanded constant. Notice that the enclosing quotation marks of the definition string are not actually part of the constant. In our example, the following string will be searched for or transmitted:

AA'' '' E_0

29.2 Constant Names

Constant names must begin with a letter or underscore character. They may include any of the following characters: underscore (_), upper or lowercase letters, and decimal numbers 0 through 9. Upper and lower case letters are distinguishable in constant names; for example, constants big, Big, and BIG will not be confused by the INTERVIEW's compiler.

29.3 Scope

The relative placement of a constant's definition within a program determines its scope, or active range.

(A) Global Constants

If you want to be able to reference a constant anywhere within a program, you must define it in the first lines of the program. Only an OBJECT or IL_BUFFERS block may precede global constants. No other block—whether a block identifier (LAYER, TEST, STATE, CONDITIONS, or ACTIONS) is entered or implied—may be placed before a global constant.

(B) Layer Constants

A layer constant must be defined before the first reference to it. The definition is placed in the lines following the LAYER identifier. (In a single–layer program, the LAYER identifier may be omitted.)

The definition of a layer constant must fall outside component blocks of the layer (outside tests and states).

A layer constant can be referenced within any test, state, or trigger which that layer contains. It may also be referenced in any other layer which follows on the spreadsheet. The only exception to this is when the constant is superseded by a constant of the same name (see the section on precedence which follows).

(C) Test Constants

A test constant must be defined at the beginning of a test block and before the first reference to the constant. While the TEST identifier may be absent in a single–layer, single–test program, the scope of a constant can only be limited to a test if it follows a TEST identifier.

A test constant cannot be defined within a state, but it can be referenced by any trigger in any state which the test contains.

29.4 Referencing Constants

Whenever you refer to a constant in your spreadsheet program, the constant name must be enclosed in double parentheses—for example, ((Frmsize)). Use the key sequence CTRL–9 and CTRL–0 to create double parentheses. Shown here is the constant ADDRESS which replaces an SDLC frame address used throughout the test. When the frame address is modified, only the constant need be changed.

```
LAYER: 2
   TEST: polling
      CONSTANTS:
       ADDRESS = "C1"
      STATE: init
         CONDITIONS: ENTER_STATE
         ACTIONS: SEND SNRM ADR= ((ADDRESS)) P/F= 1
            TIMEOUT retransm RESTART 3.000
         CONDITIONS: RCV UA ADR= ((ADDRESS)) P/F= 1
         ACTIONS: TIMEOUT retransm STOP
            RESET NR RESET NS
         NEXT_STATE: info_xfr
         CONDITIONS: TIMEOUT retransm
         NEXT_STATE: init
```

As long as syntax is observed, a constant may be used to replace a large block of text which would otherwise be repeated. Following is an example of a long, repetitive text block given as a constant definition and referenced within the program as ((LK_SETUP)). Notice that the constant definition is contained in a single logical line. The highlighted plus symbols, automatically generated by the spreadsheet editor, indicate the point at which the line wraps on the screen.

```
LAYER: 2
CONSTANTS:
LK_SETUP = "ACTIONS: SEND DISC PROMPT \"Disconnect link\" CONDI+
TIONS: RCV UA ACTIONS: PROMPT \"Disconnected\" CONDITIONS: RCV +
DISC ACTIONS: SEND UA CONDITIONS: RCV SABM ACTIONS: SEND UA PRO+
MPT \"Link restarted\""

   TEST: link_up
      STATE: fr_setup
         CONDITIONS: ENTER_STATE
         ((LK_SETUP))
      NEXT_STATE: fr
      STATE: fr
         CONDITIONS: ENTER_STATE
         ACTIONS: SEND INFO NR= 01
         CONDITIONS: RCV FRMR
         ACTIONS: PROMPT "FRMR received–test OK."
```

NOTE: Global and layer constants declared on the Protocol Spreadsheet may be referenced on any of the Trigger Setup screens as part of a receive or transmit string.

29.5 Nested Constants

The definition of a constant may include a reference to another constant. This is called nesting. An example of nested constants is shown below. On the Protocol Spreadsheet, it is possible to nest constants eight levels deep.

```
CONSTANTS:
send_lcn     = "000"
rcv_lcn      = "001"
send_data    = "data ((send_lcn)) "
send_pkt     = "send ((send_data)) "
```

NOTE: It is illegal to define two constants circularly. If, for example, you define CONSTANTS: peat = ((repeat)) and CONSTANTS: repeat = ((peat)) , you will receive an error message when you attempt to run the program.

29.6 Precedence

Programming practice usually restricts constants to a single definition. A given name should remain the same throughout the entire program.

In some special cases a constant name may have definitions that differ in separate parts of the test. It is not legal to define the same constant name twice at the same level within the same block; however, the same constant name can be defined differently inside of distinct blocks. You might, for example, define a global constant as **maxlength = "8"** at the beginning of a program. Nothing prevents you from defining a constant as **maxlength = "128"** *within* a layer or test included in the same program.

NOTE: Use the ability to give different definitions to the same constant name sparingly and with great caution.

The rule of thumb is this: When the same constant name is defined more than once, the value of the constant is controlled by the smallest block in which it resides. When that block ends, its value is controlled by the next larger block, and so on. So, a constant might have different values within a TEST, within a LAYER, and throughout the remainder of the program.

Consult the following example. Globally, the constant maxlength has a value of 8. This value holds until the constant takes on a new value in Layer 2, where it is defined as **maxlength = "128"**. Inside the Layer 2 test named shortfrm, maxlength is briefly given a value of 4. In Layer 3, the constant maxlength is not redefined, and its value returns to 8 (since this is the global definition of the constant).

```
CONSTANTS:
      maxlength = "8"
LAYER: 2
CONSTANTS:
      maxlength = "128"
  TEST: shortfrm
CONSTANTS:
         maxlength = "4"
         STATE: supfrm
            CONDITIONS:

.

.

LAYER: 3
      TEST: pktlen
         STATE: datapkts
            CONDITIONS:

.

.

.
```

29.7 Expansion

The spreadsheet editor checks constant definitions and references for several types of errors as you enter your program. In the interest of time, however, it will not expand a reference to a constant embedded in a text string. This means that nested constants are not checked for errors as you write your program.

The compiler expands these constants when you run the program, and any obvious errors will result in an operator message. Be advised, however, that it is possible for embedded constants, once expanded, to produce a valid, but unintentional, program variation.

30 Editor

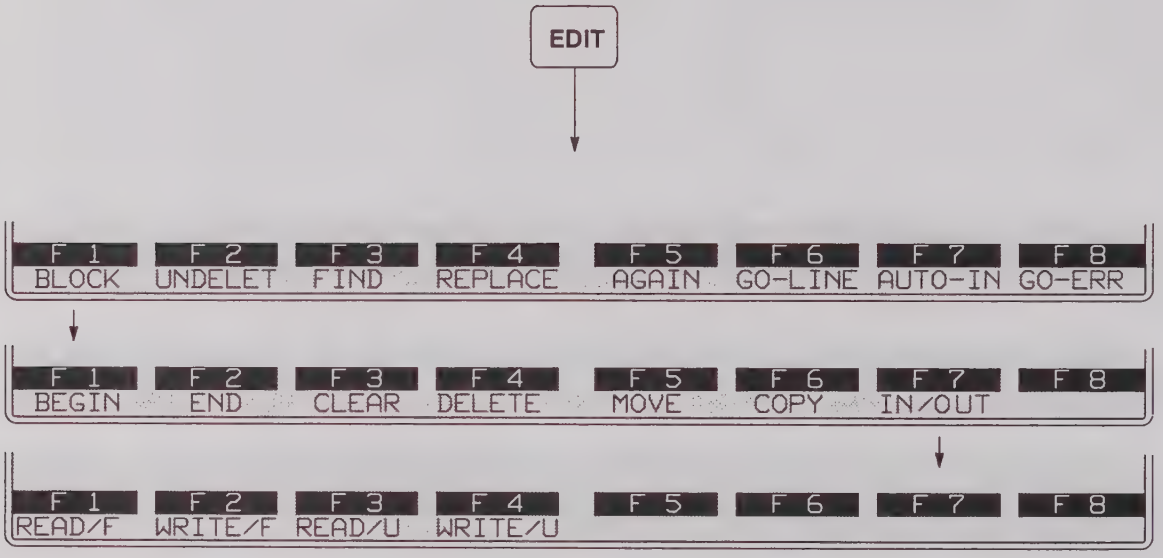


Figure 30-1 Press the *EDIT* key to access special editing functions on the Protocol Spreadsheet. Press *F1, then F7,* to access additional editing functions.

30 Editor

As you create a spreadsheet program, you may use any of the keys on the editing keypad to modify your entries. Sophisticated editing options are added to these basic functions when you press [EDIT] (see Figure 30-1). [EDIT] is an alternate action key which returns you to program function keys if you press it a second time.

30.1 Basic Editing Functions

Use the editing keypad on the right of the keyboard to perform simple editing functions (see Figure 30-2).

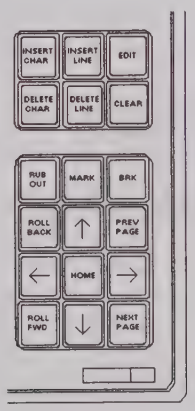


Figure 30-2 The editing keypad.

(A) Insert and Delete Keys

The top three rows of the keypad contain insert and delete functions. Available functions are insert a character, delete a character, rubout a character and insert, delete, or clear a line.

Insert Line and Delete Line functions apply to the logical line, not the physical line. A logical line has segments which end at the end of the screen but are not terminated with a [RETURN]. Instead, the logical line wraps to the next line or lines on the screen. You can distinguish a logical (wrapped) line by the highlighted plus symbols (+) at the end of each segment on the screen. When you insert a line, it appears above the first segment of the wrapped line. When you delete a logical line, all of its segments are deleted.

1. [INSERT CHAR] is an alternate action key. Press it once to enter Insert mode (the label <Insert> appears at the top left of the screen). Then type a character. The character is entered at the cursor position. All text moves right. Continue to insert characters as needed.

   Press [INSERT CHAR] again to leave Insert mode. Any character you type subsequently will overwrite an existing character at the cursor location.

2. [INSERT LINE] inserts a blank line above the logical line where the cursor is located. It also puts you into <Insert> character mode. Use [INSERT CHAR] as described in the previous paragraph to *exit* <Insert> mode.

3. [DELETE CHAR] deletes the character under the cursor. The next character to the right moves under the cursor, and remaining text shifts left.

4. [DELETE LINE] removes the logical line that the cursor is on.

5. [CLEAR] erases the remainder of the logical line from the cursor position and to the right, leaving the line empty. [SHIFT]–[CLEAR] erases the entire logical line which the cursor is on, but not the space the line occupies.

6. [RUB OUT] deletes the character just to the left of the cursor and moves the cursor left one space. Use [RUB OUT] to correct an error in the most recent keystrokes.

(B) Cursor and Movement Keys

1. [↑] and [↓] move up or down the screen one line at a time. [SHIFT]–[↑] moves the cursor to the first line of the file. [SHIFT]–[↓] moves the cursor to the last line of the file.

2. [←] and [→] move the cursor to the right or left one space at a time. [SHIFT]–[→] moves the cursor forward to the beginning of the next field. [SHIFT]–[←] moves the cursor back to the beginning of the previous field. [CTRL]–[→] moves the cursor forward to the end of the current line. [CTRL]–[←] moves the cursor back to the beginning of the current line.

3. [HOME] moves the cursor to the top left–hand corner of the current screen.

4. [ROLL BACK] leaves the cursor where it is and moves text down one line at a time.

5. [ROLL FWD] moves text up one line at a time, without changing the cursor location.

6. [PREV PAGE] recalls the previous screen of text and locates the cursor at the same relative position on the screen.

7. [NEXT PAGE] moves the cursor to the same relative position on the next screen of text.

(C) Other Keys on the Pad

1. [MARK] provides a means for "saving a place" in a program file. With the cursor at a desired location, press [MARK], then any number from 0 through 9. This marks the column and row in memory (no mark actually appears). At any time, you may locate one of ten possible marks in the file. Press [SHIFT] – [MARK], then the desired number to move forward or back through the file to the desired location.

2. [BRK] is not currently implemented.

30.2 Editing Function Keys

The editing softkeys shown in Figure 30-1 appear when you press [EDIT] with the Protocol Spreadsheet displayed. Press [EDIT] again to move back to the program function keys. When you press [F1] for BLOCK, a subset of editing options appear. Press [DONE] to move from this subset back to the top level of editing functions.

(A) Block Functions

With the regular spreadsheet programming selections displayed, press [EDIT], then BLOCK ([F1]) to display the block commands. Six editing commands (keys [F3] through [F8]) operate on blocks of text. When you are using a Clear, Delete, Move, Copy or Write command, you must mark the beginning and end of a block prior to executing the command.

1. *Begin and End.* Use BEGIN to mark the first character of a block at the cursor location. Move the cursor one position to the right of the last character you want to include in the block. Then press END. The block, once defined, is highlighted. Whenever a block is highlighted, you may clear, delete, move, or copy the block or write it to another file.

   NOTE: The block may be defined in the reverse direction. The cursor must be located one position to the right of the first character of the block and located over the last character of the block.

2. *Clear.* Press [F3] for CLEAR to "unmark" a block. The highlighting disappears to indicate that there is no longer an active editing block.

3. *Delete.* Press [F4] for DELETE to remove the marked block. Text below the block fills in the deleted area.

NOTE: You may recover a deleted block using the Undelete command on the alternate set of editing function keys. Repeated use of the Undelete command will recover up to ten deleted blocks. The text is recovered in the reverse order in which it was deleted—i.e., last deleted, first recovered.

4. *Move.* To move text, define a block and locate the cursor at the position where you want the text block to start. Then press MOVE. The text is removed from its original location and is inserted at the cursor location. The moved text remains highlighted as a block.

   To retain the original line breaks in the text, insert a blank line at the position where the new text will be located. Otherwise, inserted text will be placed at the beginning of the line marked by the cursor.

5. *Copy.* To copy text, mark a block, move the cursor to the desired location, and press COPY. A duplicate of the text block appears, highlighted. Since the block is already marked, you may copy it repeatedly without remarking it.

6. *In/Out.* To access the four Read/Write options, press the function key marked IN/OUT. A new rack of function keys appears (see Figure 30-1). These functions are explained in Section 30.2(B).

(B) Read and Write

The READ and WRITE commands are block commands but are exceptions in that they allow you to move text into and out of your program file. You can use a READ command as you would a load command to call in other Protocol Spreadsheet files. Likewise, you can save a copy of the Protocol Spreadsheet using a WRITE command.

The four command options on this rack of function keys are Read Formatted, Read Unformatted, Write Formatted, and Write Unformatted.

1. *Formatted Read and Write commands.* Read Formatted and Write Formatted are intended for use with spreadsheet files and any other files which contain nonprintable (non–ASCII) characters:

   - Special characters such as bit masks, F, G, B, A, X, «, and »
   - Any control characters outside the limited subset listed in the following paragraphs for unformatted Read and Write commands
   - "Packed" hex characters; that is, hex characters as they appear on the screen (for example $^{7}{}_{E}$, $^{3}{}_{F}$, and $^{4}{}_{0}$).

   The Write Formatted command saves these nonprintable characters as expanded ASCII and uses pound signs (#) and backslashes (\) as prefixes to mark their location for later decoding. Thus, when a file is written, # becomes ##, \ becomes \\, while $^{3}{}_{0}$ becomes #30, F becomes \7E, and so on.

The Read Formatted command decodes the expanded representations properly and displays them as they previously appeared on the Protocol Spreadsheet. If by mistake you use the Read Formatted command on a pure ASCII file which contains backslashes or pound signs, the INTERVIEW will attempt to decode the characters which immediately follow. For example, a preprocessor directive from an #include file such as

#define max 5

will be decoded as

␣ fine max 5—which obviously cannot be interpreted by the preprocessor.

2. *Unformatted Read and Write commands.* Read Unformatted and Write Unformatted are intended for use with #include files and other pure ASCII files. Any files that contain only ASCII and a limited subset of control characters may be successfully read in or written to disk with these commands. The set of control characters which are recognized and retained by these commands follows:

   - Tab (␉)
   - Form Feed (␌)
   - Carriage Return (␍)
   - Bell (␇)
   - Line Feed (␊)

   Any other control characters are stripped from the file when one of these commands is used—as are packed hex characters (␣, ␣, and so on) and special characters.

   NOTES:

   a. If you mistakenly use a Write Unformatted command on a file which contains nonprintable characters, these characters will be stripped from the file without warning.

   b. Since no messages inform you of whether file contents are formatted or unformatted when you perform a Read or Write, you should keep track of the file type for later reference. An easy way to do this is to append a suffix (such as *_u* for unformatted or *_f* for formatted) to the filename. *#include* files, which end with the *suffix .h*, require the Read Unformatted and Write Unformatted commands.

3. *How to execute a Read command.* To copy an existing file into the Protocol Spreadsheet, place the cursor at the location where you want the file to start.

Press READ/F or READ/U, whichever is appropriate (see previous paragraphs), and type in the exact filename (full or relative pathname). Then press XEQ or RETURN. The entire file is highlighted and copied at the cursor location. Any original spreadsheet text beyond the cursor position is pushed to the end of the file which has been read in.

NOTES:

a. When giving the filename to be read, provide the location of the file by disk. If the destination disk is omitted, only that one named in the current directory on the File Maintenance screen will be searched. If the file is located on the current directory disk, it will be read; otherwise, an error message will appear at the top of the screen.

b. You may read in an entire spreadsheet file without affecting the configuration of other menus in the INTERVIEW. A full program, containing the spreadsheet and the contents of all other menus, must be loaded from the File Maintenance screen. See Section 15.3(E).

4. *How to execute a Write command.* You may file a copy of all or part of your spreadsheet entries using one of the Write commands. First, mark the beginning and end of the block you wish to save to a file. Then press WRITE/F or WRITE/U, whichever is appropriate (see previous paragraphs), and give the full or relative pathname of the file when prompted. Press XEQ or RETURN. The file will appear in the directory listings on the File Maintenance screen. If you type in the name of a file which already exists, your spreadsheet text block will overwrite the entire file if the file is the *same file type*.

NOTE: If you wish to save the configuration of other menus along with your spreadsheet program, use the Save command on the File Maintenance screen; see Section 15.3(F).

(C) Other Editing Commands

To return to the main set of edit keys from the bank of Block commands, press DONE. The remaining commands in the set are described in subsequent paragraphs.

1. *Undelete.* You can return the last deleted line or block to the screen. First, locate the cursor where you want the deleted text to appear, and press UNDELET. The deleted text will be inserted at the cursor location. Repeated use of the Undelete command will recover up to ten deleted blocks.

2. *Find.* Press FIND, and the prompt "Find:", along with the cursor, appear at the top of the screen. Type in the string you wish to locate, and press RETURN. The

command performs a forward search to the end of the file. Press AGAIN to search for another occurrence of the same string. The message "Text not found" is posted at the top of the screen if the entered text does not occur between the last cursor location and the end of the file.

3. *Replace.* To replace a text string (with a maximum of 50 characters), press REPLACE. The prompt "Find:" appears at the top of the screen. Type in the string that you want to replace, and press [XEQ] or [RETURN]. The prompt "Replace with:" appears. Type in the new string, and press [XEQ] or [RETURN]. The command searches forward in the file from the cursor position and replaces the first occurrence of the string. To continue replacing the old string, press AGAIN, until the message "Text not found" is displayed at the top of the screen. The search for the text string stops at the end of the file.

   NOTES:

   a. If you want the entire file to be searched, make sure the cursor is positioned at the beginning by pressing [SHIFT]–[↑].

   b. Case does make a difference. If the string "echo" is replaced, "Echo" will not be replaced.

4. *Again.* You may repeat Find and Replace commands by executing the command, then pressing AGAIN.

5. *Go–line.* To move from one line to any other line in the file, press GO–LINE. When prompted, enter the sequential number of the line you want, and press [XEQ] or [RETURN].

6. *Auto–indent.* <Indent> will appear at the top right of the screen when Auto–indent is on. Auto–indent is an alternating function key. If the indent cue does not appear, press the function key once to turn on Auto–indent. Press the function key again to turn off indentation. Auto–indent is active both when editing keys and program function keys are active.

   NOTE: To move through the program one line at a time at the points of indentation, use the [RETURN] key instead of the [↑] and [↓] keys.

   This feature is an aid in setting up spreadsheet programs. When you use a function key to enter a keyword, the keyword appears on a new line, and, if it is a component belonging to a larger block, it is indented. For example, if you press LAYER:, the keyword is not indented, but if you press TEST, the keyword TEST: appears on a new line, indented three spaces from the first letter of its "owner"

(LAYER). When you press STATE, the keyword STATE: is indented another three spaces, to show that it is a component of the test.

NOTE: If you type in your spreadsheet entries, the last level of indentation is observed; however, other auto–indent features are not applied to manual entries.

7. *Go–error.* Most syntax errors made on the Protocol Spreadsheet are indicated by strike–through of the text where the error occurs. Press GO–ERR to move to the first editing error found moving forward (down) through the file. Press GO–ERR once more to move to the next editing error. The search for editing errors stops at the end of the file, and the message "No more errors" is displayed at the top of the screen.

   Errors which are detected by the C translator, preprocessor, or compiler are not indicated by the editor. When you press [RUN] and the test is compiled, the errors will be noted. If there are errors in the test, the INTERVIEW will revert to the Protocol Spreadsheet and display a diagnostic message about the first error rather than run the test. Press GO–ERR to search for additional errors until the "No more errors" messages is displayed.

   If you leave the Protocol Spreadsheet to go to another screen, but then want to review the list of the errors again, return to the Main Program menu. Press [F3], [EDIT], [F8] (spreadsheet screen, edit, GO–ERR). Repeat GO–ERR for the next one. When there are no more errors, a prompt to that effect will appear at the top of the screen.

   Error messages are listed in Appendix A.

31 Layer–Independent Conditions and Actions

Condition–and–action triggers are the basic programming elements on the INTERVIEW Protocol Spreadsheet. Triggers can be thought of as "If, Then" statements, organized on the spreadsheet under the headings CONDITIONS and ACTIONS. Each pairing of CONDITIONS and ACTIONS on the spreadsheet represents one trigger, similar to but also more comprehensive than one of the sixteen Trigger Setup screens (see Sections 25 and 26). Any number of triggers may be created in the spreadsheet program.

During a test, a trigger condition is active (potentially true) whenever the state it belongs to is active. An action is taken whenever the condition (or set of conditions) preceding it is true.

This section covers those conditions and actions that are *not* local to a particular protocol at a particular layer of programming. These are the conditions and actions that are made available as softkey selections in every state in the program without exception.

31.1 Naming Requirements

Flags, accumulators, signals, counters, timers, and timeouts are layer–independent trigger entities that are created by the user in any number and combination and called out by *keyword* (FLAG, ACCUMULATE, SIGNAL, COUNTER, TIMER, TIMEOUT) and by *name*. The names are assigned by the user and referenced in triggers throughout the program.

A name on the Protocol Spreadsheet must not exceed sixteen characters nor include any except the fifty–two alpha characters (upper and lower cases) and the ten numeric characters in addition to the underscore (_) character. The first character in each name must be an alpha character.

The practical size limit for the names of counters, timers, and accumulators is eight characters, since a longer name cannot be called out on the tabular and graphic statistics screens.

For the sake of program readability, we recommend that all user–assigned names be entered in lower case. In this way they will be distinguishable from keywords. The spreadsheet compiler does not insist on lower case for user–assigned names, however.

The spreadsheet compiler does treat upper– and lower–case names as distinct. A timer named **delay** will not be referenced by the name **DELAY** (or **Delay**), for example. Keywords are treated differently: typing timer has the same effect as typing Timer or TIMER or pressing the softkey that writes TIMER to the screen.

Names of different entities need not be kept distinct. The program will have no trouble keeping a SIGNAL named **ready** separate from a FLAG of the same name. (The user may have difficulty keeping them separate, however.)

31.2 Rules for Combining Conditions

Several layer–independent conditions are "transitional" (or "instantaneous") conditions, in that they are true only for the *instant* that they transition to true. These transitional conditions are enter–state, timeout, keyboard, time–of–day, and signal conditions. Triggers that combine two transitional conditions are illegal and will not compile, since there is no chance of two transitional events occurring simultaneously.

The other class of layer–independent conditions, comprised of buffer–full, counter, and flag conditions, may be thought of as transitional/status. When used alone in a trigger, these conditions are true only at the moment they transition to true.

For example, the condition COUNTER retries GE 5, used by itself preceding an Actions block, will be true once when the counter increments from 4 to 5, but not when the same counter increments to 6. For the condition ever to be true again, the counter must first transition to a value less than 5.

When used in combination with transitional conditions, these transitional/status conditions are checked for a current *status* of true at the moment the transitional condition transitions true. They may retain this status of true indefinitely.

Here is an example of a transitional/status condition (counter) used in combination with a transitional condition (timeout).

```
CONDITIONS: TIMEOUT response
   COUNTER retries GE 5
ACTIONS: ALARM
```

This set of conditions will be true every time the timeout occurs as long as the counter retains a *status* of greater than or equal to 5.

When a transitional/status condition is used in combination with one or more other transitional/status conditions, the first condition in the user–defined sequence of conditions will be transitional, while the others will be checked for truth or falsity only when the first condition transitions to true. Take, for example, a scenario where a counter increments five times and then a flag increments five times. On the fifth flag increment, the following set of conditions will be true:

```
CONDITIONS: FLAG true_last 101
   COUNTER true_first EQ 5
```

The conditions are satisfied because the flag is transitional while the counter is static: at the moment the flag transitions to binary 101 (decimal 5), the counter is checked for a status of 5. Both are true. But given the same scenario, this set of conditions is false:

```
CONDITIONS: COUNTER true_first EQ 5
   FLAG true_last 101
```

Here, the counter condition is transitional, the flag is static—simply because the counter condition is listed first. The flag condition is checked only at the moment the counter attains the count of 5. After that, the flag is not checked again.

The condition logic is streamlined in this manner in order to be economical of processor time, on the assumption that in a typical application the user knows which of two conditions will be satisfied first. If users do not know whether the counter or the flag in the above example will increment to 5 first, nothing prevents them from entering two triggers, both having the same conditions but in a different sequence. Or they may enter the pair of conditions on a Trigger Setup menu, where combined transitional/status conditions generate enough code to cover all contingencies. See Section 25.2(B)2.

> **NOTE:** Additional rules may apply when the COUNTER or FLAG transitional/status condition is used in a spreadsheet program compiled and saved as a linkable–object file. See Section 28.4(B).

31.3 Layer–Independent Conditions

The eight softkeys that represent the full set of layer–independent conditions are shown in Figure 31-1.

(A) Enter State

This condition is true immediately as the current state is entered. Control of the action in effect reverts to the previous state. In the example below, ENTER_STATE is used as the condition for an alarm action in **second** state. The counter condition in **first** state effectively controls this alarm.

```
STATE: first
   CONDITIONS: COUNTER frm_err EQ 10
   NEXT STATE: second
STATE: second
   CONDITIONS: ENTER_STATE
   ACTIONS: ALARM
```

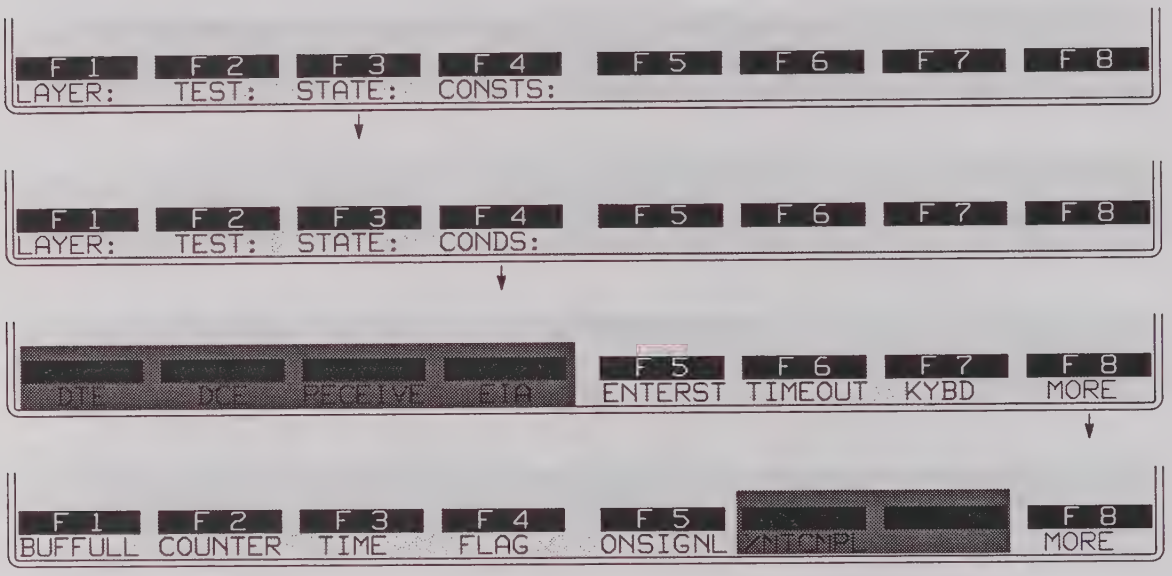


Figure 31-1 The eight layer–independent conditions are shown in the bottom two racks of softkeys.

(B) Timeout

Any number of decrementing timeout timers may be started as trigger actions and monitored by trigger conditions. The condition is true when the timeout timer expires.

Here is an example of a timeout condition:

```
TIMEOUT response
```

where **response** is the name of the timeout timer.

After pressing the TIMEOUT softkey or typing TIMEOUT followed by space, enter a name. The name can reference a timeout timer that was started either in a spreadsheet action or a trigger–menu action.

(C) Keyboard

Enter a list of characters produced by keystrokes. Any key or key–combination that produces a character on the ASCII table in Appendix D1 is valid input in this field. Lists in the spreadsheet program can extend to 128 characters.

In Run mode when any key on the list is pressed, the condition will be true and (if this is the only condition) will initiate a trigger action.

An example of a keyboard condition is the following:

```
CONDITIONS: KEYBOARD "1 "
```

Note the space following the 1 entry. Here the 1 key *or the space bar* will satisfy the trigger condition. Dual quotation marks are required for all lists and strings on the Protocol Spreadsheet.

(D) Buffer Full

This condition is true at the moment the 64–Kbyte character buffer is full. Use this condition to trigger a display–freeze (CAPTURE BOTH OFF) whenever the earliest data in the display buffer is the most important and you do not want it to be overwritten. Here is an example of a trigger that will retain the first full buffer of data:

```
CONDITIONS: BUFFER_FULL
ACTIONS: CAPTURE BOTH OFF
```

(E) Counter

Any counter named and operated as a trigger action may be monitored as a trigger condition. To create a counter condition, press the COUNTER softkey or type COUNTER followed by a space.

> **NOTE:** A counter named on a Trigger Menu screen also refers to a spreadsheet counter as long as the name matches. Timeouts and timers can also be shared between the Trigger Menu screens and the spreadsheet.

> **NOTE ALSO:** Trigger Setup screens monitor counter values from 0 to 999,999. However, Protocol Spreadsheet triggers can monitor counter values up to 4,294,967,295.

The following is an example of a spreadsheet counter condition:

```
CONDITIONS: COUNTER byte_no EQ 128
```

where **byte_no** is the name, EQ(ual) is the relational operator, and **128** is the decimal value.

1. *Enter counter name.* Name the counter to be monitored. See Section 31.1, Naming Requirements.

2. *Relational operator.* As soon as a counter name has been typed and followed by a space, a rack of softkeys appears with names of relational operators. See Figure 31-2.

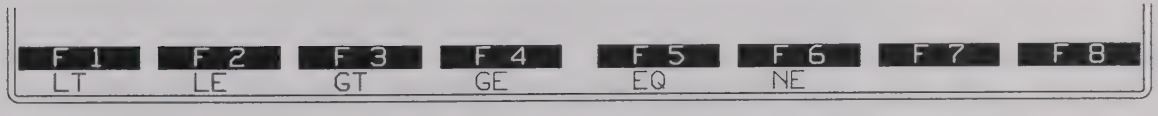


Figure 31-2 A set of relational operators compares the counter value to a user–entered value.

Make the appropriate selection to specify when the counter condition will be true. The counter may be tested for a value equal to (EQ), not equal to (NE), greater than or equal to (GE), less than or equal to (LE), strictly greater than (GT), or strictly less than (LT) the value entered on the spreadsheet.

When a COUNTER condition is used alone, it is a *transitional* condition. This means that it is true only when it transitions to true. For example, a condition that said COUNTER **drops** NE 5 would be true when COUNTER **drops** transitioned from 5 to 6—that is, on the transition from *equal 5* to *not equal 5*; but the condition would not be true when 6 changed to 7.

In combination with another condition (that is, more than one condition *per* action or set of actions), a COUNTER condition normally is a status condition, not a transitional condition. As a status condition, COUNTER **drops** NE 5 is true any time the status of the counter is not 5. Refer to Section 31.2, Rules for Combining Conditions.

> **NOTE:** Additional rules may apply when the COUNTER transitional/status condition is used in a spreadsheet program compiled and saved as a linkable–object file. See Section 28.4(B).

3. *Enter the counter value.* Enter the value as a whole decimal number. Each condition can monitor a 32–bit counter for decimal values ranging from 0 to 4,294,967,295.

> **NOTE:** The Current value for a counter on the Tabular Statistics screen is maintained to seven decimal places, for a maximum counter display of 9,999,999. The 32–bit binary counter can attain much higher values than this, however—the decimal display on the statistics screen merely rolls over to zero and continues counting. Spreadsheet counter conditions can monitor for values up to the maximum of over four billion. If a trigger looks for a counter value higher than this maximum, it will never be satisfied.

(F) Time

The time of day once a day or once a month can satisfy a trigger condition. Here, for example, is a trigger condition that comes true at 3 P.M. each day:

```
CONDITIONS: TIME 1500
```

1. *Enter day of month or time of day.* Press the TIME softkey or type TIME followed by a space. The next entry will signify day of month if it is a two–digit entry. If it is four digits, it will signify the time of day in twenty–four hour format.

2. *Enter time of day.* If the entry following TIME is a two–digit, day–of–month entry, it must be followed by time of day in a four–digit, twenty–four hour format.

(G) Flag

Sixteen internal flag bits are reserved for every flag mask that is named in Protocol Spreadsheet conditions and actions.

> **NOTE:** The eight flag bits on the Trigger Setup screens are the low–order bits of a flag mask that can be accessed on the Protocol Spreadsheet by the name **trig_flag**.

A flag condition still is valid when fewer than sixteen flag bits are specified. The flag values that are specified are right–justified when the program is compiled, and leading X's (don't cares) are assumed.

The internal flag normally is a static condition when it is used in combination with other trigger conditions—that is, more than one condition per action or set of actions. Refer to Section 31.2, Rules for Combining Conditions. Since flag bits are completely under program control and can be used in combination with other conditions, they are useful chiefly to enable or disable entire triggers.

> **NOTE:** Additional rules may apply when the FLAG transitional/status condition is used in a spreadsheet program compiled and saved as a linkable–object file. See Section 28.4(B).

For example, a trigger action is taken if a flag bit is 1 and a % character is seen. Setting the flag to zero effectively disables this trigger.

The following is an example of a flag condition :

```
CONDITIONS: FLAG nak 1X
```

where nak is the name of the flag and XXXXXXXXXXXXXX1X is the flag bit mask.

1. *Enter the flag name.* After pressing the FLAG softkey or typing FLAG followed by space, enter a name not exceeding eight characters, beginning with an alpha character.

2. *Enter the flag condition bit mask.* A flag mask follows the flag name. The mask can include up to sixteen bits (with no spaces between them). Since the number of flag masks in your program is unlimited, you may want to restrict your masks to one or two bits. In effect you will be giving each bit or pair of bits a name.

   Legal bit–entries are 1, 0, or X (for "don't care"). Press [X] or [DON'T CARE] to enter an X. The condition will not test this bit.

(H) On Signal

Signals are communicated between tests and between layers. They are the simplest way to use an event in one test to start a state or an action in another test. Here is an example of an on–signal condition:

```
ON_SIGNAL testfail
```

After pushing the ONSIGNL softkey or typing ON_SIGNAL followed by space, enter the name of a signal you have created (or intend to create) in a trigger action.

31.4 Layer–Independent Actions

When a block of conditions has been entered, press [DONE] to access the ACTIONS softkey. The actions that are available in all states without exception are shown in Figure 31-3 as they appear in three successive racks of softkeys.

(A) Counter

The Protocol Spreadsheet screen can control any number of counters. The Tabular Statistics screen is an expanding display that can provide statistics for 100 counters, timers, and accumulators.

Here is an example of a counter action:

```
ACTIONS: COUNTER datapaks INC
```

1. *Enter counter name.* A counter can be unique to one trigger action or it may be shared with other actions and other triggers, which can monitor it and change its values. As long as the same counter name is used, the same counter is invoked.

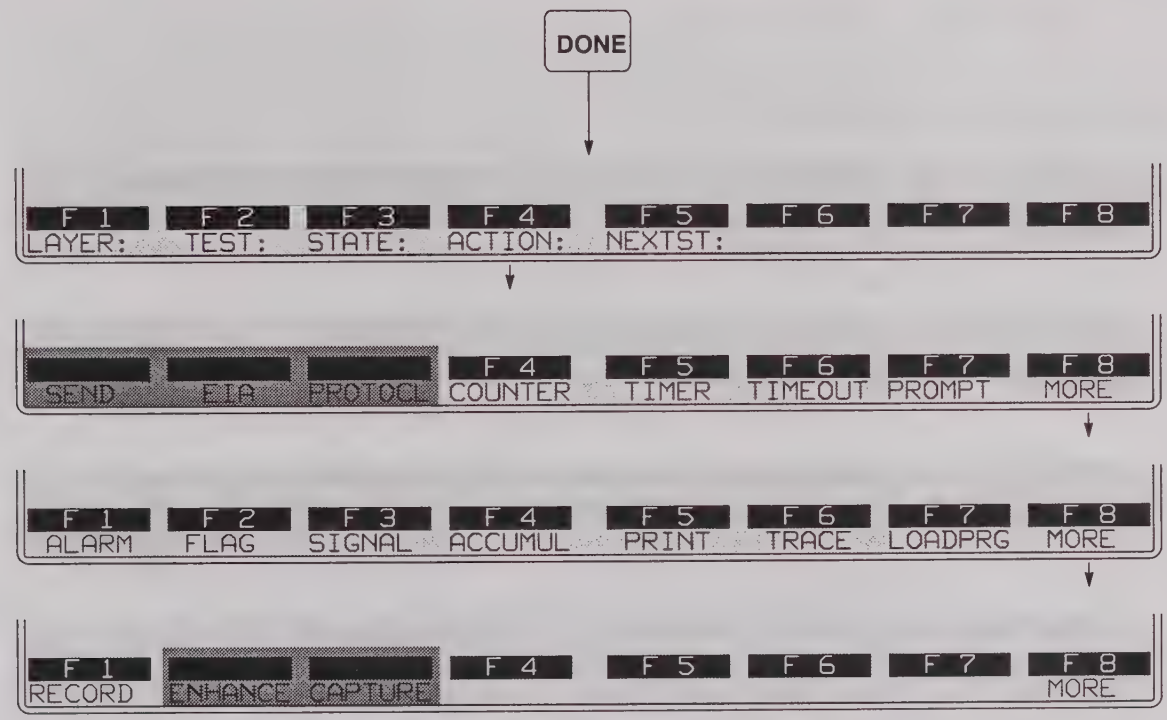


Figure 31-3 The twelve layer–independent actions are spread over the bottom three racks of softkeys in this figure.

NOTE: A counter named on a Trigger Menu screen also refers to a spreadsheet counter as long as the name matches. Timeouts and timers can also be shared between the Trigger Menu screens and the spreadsheet.

NOTE ALSO: Trigger Setup screens monitor counter values from 0 to 999,999. However, Protocol Spreadsheet triggers can monitor counter values up to 4,294,967,295.

After naming the counter, select among the actions shown in the rack of softkeys in Figure 31-4.

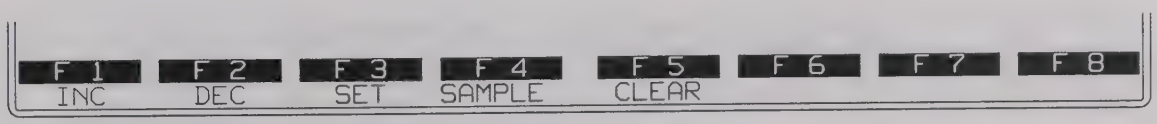


Figure 31-4 Counter actions.

2. *Increment.* Thirty–two bits are reserved for each counter. Therefore a counter will roll over after it attains a decimal value of 4,294,967,295. Spreadsheet

conditions can monitor a counter for any value from zero to the maximum. (Trigger Menu conditions can monitor up to a count of 999,999.) Note, however, that the counter value will only be displayed up to seven decimal places on the Tabular Statistics screen. The maximum displayed value therefore is 9,999,999.

3. *Decrement.* When this action is selected, each trigger occurrence subtracts 1 from the counter. A counter that decrements below zero wraps to 4,294,967,295. The last seven decimal places of this maximum value will be displayed in the **Current** column on the Tabular Statistics screen.

4. *Set.* Select SET in order to specify the value that the counter will take when the trigger comes true. Enter a decimal value for the counter. To reset a counter without taking a statistical sample, use the SET action and enter a value of zero.

5. *Sample.* This action stores the current value of the counter and then resets it to zero. The stored value is posted immediately to the statistics display in the **Last** column. This value is compared with previous **Last** values in order to compute **Minimum**, **Maximum**, and **Average** values for statistical display. Refer to Section 21 for a discussion of tabular statistics.

6. *Clear.* This action resets the counter to zero and also resets last, minimum, maximum, and average values for the counter.

(B) Timer

The Protocol Spreadsheet can control any number of timers. The Tabular Statistics screen is an expanding display that can provide statistics for 100 counters, timers, and accumulators.

While timers can be run and sampled as trigger actions, they are not available as trigger conditions. Timeouts, not timers, are the mechanism that allows you to trigger off of elapsed time.

Timer values may be based on an internal "wall" clock, or, if time ticks are enabled on the Front–End Buffer menu screen, on ticks that are stored along with the data. The "tick" mode of timing is the most accurate, especially when data is played back and you do not want playback conditions such as speed and idle–suppression to affect the timers.

Here is an example of a timer action:

```
ACTIONS: TIMER session SAMPLE
```

1. *Enter timer name.* After pressing the TIMER softkey or typing TIMER followed by a space, enter a name. Like counters and timeouts, a timer can be shared between the spreadsheet program and the Trigger Menu screens. If the same name is used, the same timer is invoked.

After naming a timer, select among the actions shown on softkeys in Figure 31-5.

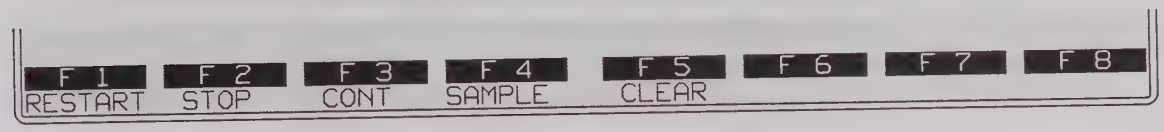


Figure 31-5 Timer actions.

2. *Restart*. Use RESTART to start a timer. This causes the timer to reset to zero and begin incrementing. A restart does not affect any statistical values except **Current**.

3. *Stop*. A stop action suspends the timer at its present value. The timer may be started again at this value by a Continue action on another trigger.

4. *Continue*. This action restarts the timer beginning at the value that was frozen in the **Current** column when the timer was stopped. The Continue action has no effect on a timer that is incrementing already.

5. *Sample*. Sampling a timer resets it at zero and stops it. Prior to resetting, the current value is posted as a **Last** value and passed along for other statistical tabulation.

6. *Clear*. Clearing a timer resets and stops the timer and clears the last, minimum, maximum, and average values.

(C) Timeout

Any number of timeouts can be started and stopped in the spreadsheet program. Timeouts are timers that count down instead of up. Their values are not read on any statistical display; but when they time down to zero, they satisfy trigger conditions that monitor them by name. Timeout timers that are named on the Protocol Spreadsheet also may be monitored and controlled on the Trigger Menu screens.

Here is an example of a timeout action:

```
ACTIONS: TIMEOUT t2 RESTART 3
```

where **t2** is the name of the timeout and **3** is its duration in seconds.

1. *Enter timeout name*. After pressing the TIMEOUT softkey or typing TIMEOUT followed by space, enter the name of the timeout. As soon as a name has been entered and followed by a space, a rack of softkeys appears with the names of two timeout actions, RESTART and STOP.

2. *Restart*. Select RESTART to start the timer running down.

3. *Stop*. Select STOP to halt the timer and prevent the timeout.

4. *Enter timeout value*. The duration of the timeout is entered in seconds. The timeout value is a decimal field in which entries are valid to the millisecond (0.001). For values under 1 second, you must precede the decimal with a leading zero, as follows:

```
TIMEOUT delay RESTART 0.25
```

The maximum timeout entry in this field is 65.535 seconds.

You may expand the maximum timeout with a program such as the following, which produces an alarm every twenty minutes.

```
STATE: twenty_min_alarm
   CONDITIONS: ENTER_STATE
   ACTIONS: TIMEOUT sixtysec RESTART 60
   CONDITIONS: TIMEOUT sixtysec
   ACTIONS: COUNTER minutes INC
            TIMEOUT sixtysec RESTART 60
   CONDITIONS: COUNTER minutes EQ 20
   ACTIONS: COUNTER minutes SET 0
            ALARM
```

(D) Prompt

Prompts are user–entered ASCII messages that appear on the second status line at the top of the screen in Run mode as a result of a trigger coming true. They are messages to operators, from the program, alerting them to important protocol or program events. Prompts are written to the second status line of any current Run–mode display screen. Switching to Freeze mode or to another display screen clears the prompt from all screens except the Display Window.

NOTE: The prompt line is not zeroed out with each new prompt, and prompts are overwritten only to the extent of the new prompt. For example, the prompt "POLL" does not completely overwrite the prompt "SELECT"—the result will be "POLLCT." It is a good practice to establish a uniform prompt length and space–fill shorter prompts to that length.

Special C functions that position the cursor anywhere on the prompt line (or elsewhere in the display) and write messages to the cursor position are discussed in Section 60.

A prompt that has been triggered in Run mode is illustrated in Figure 31-6. Here is the same prompt as it appears on the Protocol Spreadsheet:

```
ACTIONS: PROMPT "Echoed message received"
```

Backslashes and double–quotation marks may be included in prompt messages if they are preceded by backslashes, in accordance with the rules for entering these characters in transmit strings. See Table 33-2. Example:

```
ACTIONS: PROMPT " \" hello\" string received"
```

1. *Enter prompt message*. After pressing the PROMPT softkey or typing PROMPT followed by a space, enter a message in quotation marks. The message should not exceed 64 characters, the width of the screen.

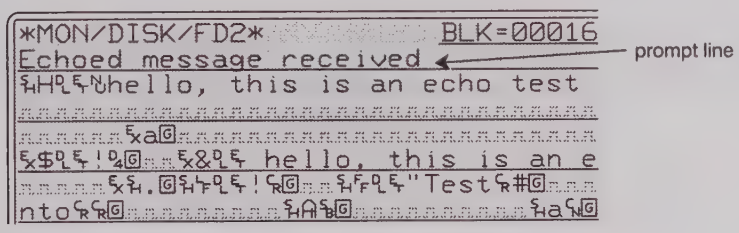


Figure 31-6 User–defined prompts are displayed at the top of the Run screen.

(E) Alarm

The alarm is a tone of less than a second duration. The alarm is sounded each time the trigger comes true. If the tone lasts longer than a second, the alarm has been triggered more than once.

The alarm action on the spreadsheet is simply the word ALARM.

(F) Flag

Internal flags are special programming bits. They can be set on or off by triggers and sensed by triggers. Flags come in masks of up to sixteen bits. Each flag mask is named and referenced by the spreadsheet program.

Any number of flag masks may be created. Flags are common to all tests and layers: if a flag name is used in tests in two different layers, it refers to the same sixteen programming bits.

A flag action still is valid when fewer than sixteen flag bits are specified. The flag values that are specified are right–justified when program is compiled, and leading X's (don't cares) are assumed.

> **NOTE:** The eight flag bits on the Trigger Setup screens are the low–order bits of a flag mask that can be accessed on the Protocol Spreadsheet by the name **trig_flag**.

Here is an example of a flag action:

```
ACTIONS: FLAG nak SET 0X
```

where nak is the name of the flag, SET is the action, and **0** is the only bit in the mask affected by the set action.

1. *Enter the flag name.* After pressing the softkey for FLAG, or typing FLAG followed by a space, a rack of softkeys appears with the names of flag actions. See Figure 31-7.

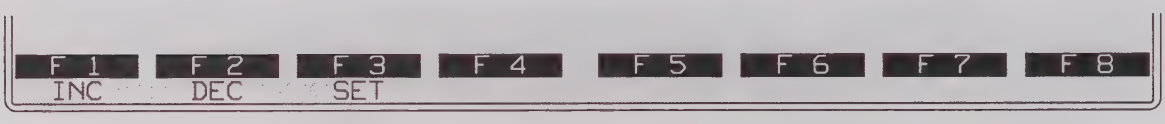


Figure 31-7 Flag actions.

2. *Increment.* The mask can be used as a 16–bit binary counter. The INC action increases the value of the mask by one each time the trigger is true.

   As the mask increments above 65,535, it wraps to zero.

   The INC action always toggles the least significant flag bit. If you monitor the flag for only one bit (for example, FLAG flagname 1), the INC action will toggle the condition true/false. This can be a useful tool when you want *every second occurrence* of an event to trigger an action.

3. *Decrement.* This action decreases the value of the flag byte by one each time the trigger is true. When the mask decrements below zero, it wraps to 65,535.

4. *Set.* This action rewrites the flag bits according to the flag–action bit mask that you enter following the SET keyword. The bit mask is comprised of up to sixteen 0's, 1's, and X's (no change).

   When you enter fewer than 16 bits, you are leaving the leftmost bits in the mask unspecified. The action will not change the condition of unspecified bits.

(G) Signal

Use signals to convey instructions to other tests and other layers where conditions are monitoring these signals by name.

Other internal programming mechanisms, such as flags and counters, are common to all tests and layers and may perform a signalling function. Signals, however, are more efficient in that they are reusable: a signal that is sent and monitored can be sent and monitored again ten seconds later, but an action that sets a flag to 1 cannot be used again until another action has intervened to reset the flag to zero.

After pressing the SIGNAL softkey or typing SIGNAL followed by space, enter the name of the signal. Often the name will be descriptive of the event being signalled. The following is an example of a signal action:

```
ACTIONS: SIGNAL testfail
```

(H) Accumulate

The accumulate action reads a specified value for a counter or timer and presents this value to tabular and graphic statistics screens for statistical breakout. This action is distinct from the sampling action of a counter or timer in this important respect: sampling a counter or timer also resets it to zero. Accumulating a counter or timer has no effect on the ongoing counting or timing function. Examples of accumulators are given in Section 21, Tabular Statistics.

Values for more than one counter or timer may be brought into a single accumulator. For example, separate timers might measure response times for a group of multidropped DTEs. At the end of the test, a value for each timer could be brought, in separate trigger actions, into one accumulator.

Each accumulate action specifies one value only for a counter or timer. Thus, the accumulator might provide meaningful statistical data based, for example, on maximum values only for a group of timers.

Here is an example of an accumulate action:

ACTIONS: ACCUMULATE alldrop COUNTER badbcc_a LAST

where **alldrop** is the name of the accumulator, **badbcc_a** is the name of a counter, and it is the last value of the counter that is being accumulated.

1. *Enter the accumulator name.* Both the accumulator and one counter or timer are referenced in the accumulate action. Counters and timers are referenced, not created, in accumulate actions.

   An accumulator is created by being named in an accumulate action. Like counters and timers, accumulators can be given display lines on either or both of the statistics screens.

2. *Clear.* This action clears the last, minimum, maximum, and average values of the accumulator. (Since accumulators neither count nor time, they never display a current value.)

3. *Counter.* This action accumulates a value for the counter named immediately following the keyword COUNTER. After the counter is named, one value for that counter is selected from the rack of softkeys in Figure 31-8.

4. *Timer.* This action accumulates a value for the timer named immediately following the keyword TIMER. After the timer is named, one value for that timer is selected from the rack of softkeys in Figure 31-8.

Figure 31-8 Counters and timers are accumulated with respect to one statistical value only.

(l) Print

Time–stamped printouts of single lines of data can be commanded by the spreadsheet program. The data can be a line of tabular statistics for an accumulator, counter, or timer; or a user–prompt that is sent to the printer after it has been written to the second line of the screen.

The following is an example of a print action:

```
ACTIONS: PRINT TIMER echotime  MILLISECONDS
```

After pressing the softkey for PRINT or typing PRINT followed by a space, select an option for the type of data to be printed from the new rack of softkeys shown in Figure 31-9.

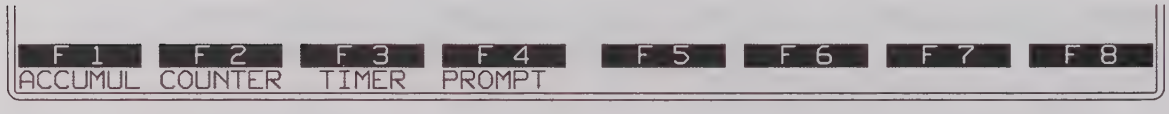


Figure 31-9 Four types of data may be printed out as a trigger action.

1. *Accumulator.* When this action is taken, the line of tabular statistics for the accumulator that you name will be printed. A line of tabular statistics includes last, minimum, maximum, and average values for an accumulator. Since accumulators neither count nor time, they never display a current value.

2. *Counter.* When this action is taken, the line of tabular statistics for the counter that you name will be printed. A line of tabular statistics includes current, last, minimum, maximum, and average values for a counter.

3. *Timer.* After the timer is named, a timer rate is selected from a new rack of softkeys as shown in Figure 31-10.

Figure 31-10 After a timer is named for printout display, a new softkey rack allows you to specify unit of time.

The selected rate will only display values to the smallest place value afforded by the tick rate selected on the FEB Setup screen. For example, if *milli*seconds is selected on the FEB screen, choosing *micro*seconds on the print–timer softkey selection will simply display three additional zeros as place holders—it will not calculate a more precise reading. Thus, the most accurate selection for this example would be milliseconds, matching the FEB selection.

When a timer is controlled by a nondata event such as a keyboard condition, it will reference a "wall–time" clock whose smallest resolution is a millisecond.

When the PRINT TIMER action is taken, the line of tabular statistics for the timer that you name will be printed. A line of tabular statistics includes current, last, minimum, maximum, and average values for a timer. Figure 31-11 is an example of such a printout for the program given below.

```
STATE: message
   CONDITIONS: DTE STRING "hello"
   ACTIONS: PROMPT "String sent by DTE"
      TIMER echotime RESTART
   NEXT_STATE: echo
STATE: echo
   CONDITIONS: DCE STRING "hello"
   ACTIONS: PROMPT "Same string by DCE"
      TIMER echotime STOP
      TIMER echotime SAMPLE
      PRINT TIMER echotime MILLISECONDS
   NEXT_STATE: message
```

```
Time          Name     Current  Last  Minimum  Maximum  Average  Unit

09/29 16:13   echotime     0    452    452      452    452.00   MSECS
09/29 16:13   echotime     0    341    341      452    396.50   MSECS
09/29 16:13   echotime     0    428    341      452    407.00   MSECS
```

Figure 31-11 In this printout, a PRINT TIMER action has been triggered three times.

4. *Prompt.* The PRINT PROMPT action is designed to be added to an action block that already contains a prompt. The example below inserts PRINT PROMPT actions into the program described in the previous section. The user does not have to key in a long prompt message twice, once for the printout and once for the screen. The printout for this program is shown in Figure 31-12.

```
STATE: message
   CONDITIONS: DTE STRING "hello"
   ACTIONS: PROMPT "String sent by DTE"
      PRINT PROMPT
      TIMER echotime RESTART
   NEXT_STATE: echo
STATE: echo
   CONDITIONS: DCE STRING "hello"
   ACTIONS: PROMPT "Same string by DCE"
      TIMER echotime STOP
      PRINT PROMPT
      TIMER echotime SAMPLE
      PRINT TIMER echotime MILLISECONDS
   NEXT_STATE: message
```

```
Time      Name      Current  Last Minimum Maximum   Average Unit

09/29 16:13 String sent by DTE
09/29 16:13 Same string by DCE
09/29 16:13 echotime      0   452   452   452   452.00 MSECS
09/29 16:13 String sent by DTE
09/29 16:13 Same string by DCE
09/29 16:13 echotime      0   341   341   452   396.50 MSECS
09/29 16:13 String sent by DTE
09/29 16:13 Same string by DCE
09/29 16:13 echotime      0   428   341   452   407.00 MSECS
```

Figure 31-12 Printout resulting from a combination of PRINT PROMPT and PRINT TIMER actions.

NOTE: If you want to print multiple prompts, place each PROMPT and PRINT PROMPT pair in its own conditions/actions block. (Otherwise, only one prompt will be printed since prompts overwrite each other.)

(J) Trace

Traces are user–entered ASCII data strings, identical to prompts in all ways except in their mode of display: traces are posted one to a line in the multiline Program Trace display (see Section 6.6), while prompts appear on the second status line in all data–display modes (including the Program Trace).

Numerous layers and numerous tests per layer can be active concurrently in the INTERVIEW. The Program Trace can be set up to track state–to–state movement only in a particular **Layer** and **Test** identified by the operator on the Display Setup menu. State names can be included in the Program Trace via the **Display States:** YES selection on the the Display Setup menu. See Figure 31-13.

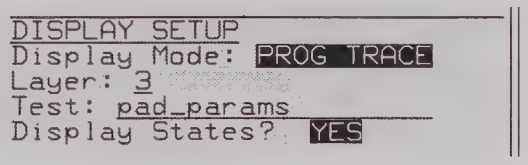


Figure 31-13 The user may select a particular layer and test for a Program Trace.

Traces are debugging tools. Inside a dead–end state they can inform you whether a particular condition that you are expecting is coming true. Prompts, by contrast, have a much fainter "trail": it is hard to be certain that a prompt was not activated and then overwritten by another prompt.

Traces also allow you to keep a record of selected protocol events—to design your own protocol analysis. Since they are written to consecutive lines rather than overwritten by other traces, they are highly useful when you are trying to track protocol events that occur in quick succession.

The following is an example of a trace action:

```
ACTIONS: TRACE " Network congestion"
```

1. *Enter trace message.* After pressing the TRACE softkey or typing TRACE followed by a space, enter a message in quotation marks.

(K) Load Program

A program (source code or object code) or setup that is stored in a file on hard disk or on a disk in either of the microfloppy drives can be loaded in by trigger action. This Load Program function is a means of chaining tests together.

Program files are a full set of configured menus, including the Layer Setup screen, Trigger Setup screens, and the Protocol Spreadsheet. Object files are the precompiled object–code versions of programs. Setup files are a set of configured menus which excludes trigger setups, the Layer Setup screen, and the spreadsheet. Remember that loading a program or setup file overwrites the program or setup file already in memory. Loading an object file overwrites only the object code of whatever program (if any) was compiled most recently. The new object file will not affect the data on any setup menu or programming screen.

EIA statuses can be maintained in between programs by a special menu selection on the Interface Control menu screen. (See Section 12.)

The following is an example of a Load Program action:

```
ACTIONS: LOAD_PROGRAM "FD1/usr/sna/sna_bind"
```

where *FD1* is microfloppy–diskette drive 1, the first slash (/) is the root directory, *usr* is the highest level of user–created files, *sna* is another directory, and *sna_bind* is the filename.

1. *Enter program name.* Enter the absolute pathname of your file. Put the name in quotation marks.

(L) Record

Use the RECORD action to activate or suspend line–data recording or disk–data playback. When the Line Setup menu is configured to monitor a disk, RECORD controls playback; otherwise it applies to recording. There are two selections under RECORD. Select ON to activate, or OFF to suspend, recording or playback.

During recording, the top status line of Run–mode screens will show incrementing block numbers and an "R" displayed in the record/playback field. During playback, a "P" is displayed. Whenever recording or playback has been suspended, an "S" is displayed.

For data playback, the status field will be blank if a disk is not present in the selected drive or when the end of the data–acquisition tracks are reached. This field will also be blank if you enter a starting block number on the Line Setup menu that a) precedes the block number at which data actually begins, or b) exceeds the block number at which data actually ends. Change your entry to zero.

For data recording, the status field will be blank when the end of RAM or the data–acquisition tracks is reached. It will also be blank if the **Capture Memory** field indicates that you will record to disk, but no disk is present in the selected drive or data–acquisition tracks are not available on the disk.

32 Layer 1 Conditions and Actions

There are seven protocol layers in the OSI (Open Systems Interconnect) model that is adopted in the INTERVIEW 8000 Series. Each layer reserves a distinctive set of trigger conditions and actions on the Protocol Spreadsheet.

As a rule, spreadsheet components for a given layer are loaded from disk via the Layer Setup screen. Layer 1, the Physical Layer, is an exception to this rule. Layer 1 conditions and actions are enabled on the Protocol Spreadsheet when the unit powers up.

Depending on the Test Interface Module (TIM) installed in the unit, the power−up also enables an Interface Control Menu screen, different for each module, that controls many Layer 1 parameters. For this reason, the set of Layer 1 conditions and actions is relatively small. For units equipped with a dual−port TIM, the selections are slightly different from units using a single−port TIM.

32.1 Single−Port Layer 1 Conditions

To bring up the bank of softkey conditions for Layer 1 using a single−port TIM, first press the CONDITIONS softkey. This key becomes available when the cursor enters a programming block at the state level.

The first four condition softkeys—DTE, DCE, RECEIVE, and EIA—belong to Layer 1. These are followed by generic conditions discussed in Section 31. The set of Layer 1 conditions is shown in Figure 32-1. The softkey for a fifth Layer 1 condition—XMIT_COMPLETE—appears on the second rack of condition softkeys shown at the bottom of the figure.

EIA is a transitional/status condition and may be combined with other conditions. The other Layer 1 conditions are transitional only. Refer to Section 31.2 for a discussion of how conditions may be combined.

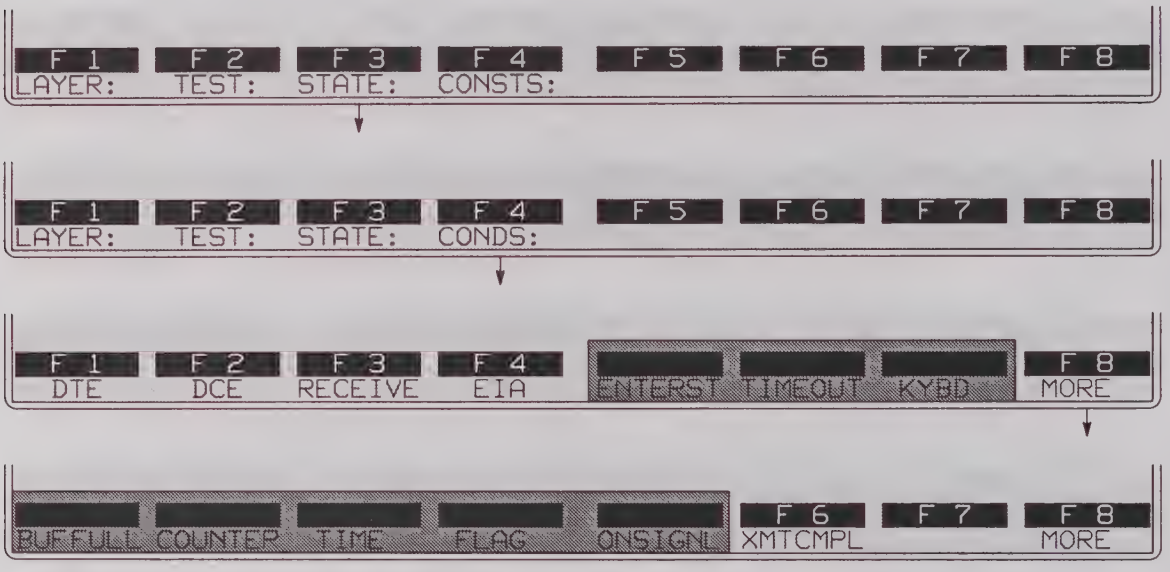


Figure 32-1 Layer 1 conditions.

(A) Data

The first three trigger conditions at Layer 1 are valid when using a single–port TIM and for Port 1 when using a dual–port TIM (Port 2 is discussed in Section 32.2). They can monitor one of the two data leads for a specific data event. This event can be any of several characters, a string of characters, a good BCC following the character or string, an error revealed by a block or parity check, and so on.

Data conditions at Layer 1 monitor the entire data stream (for Port 1 when using a dual–port TIM). Conditions in other layers also check the data leads, of course, but conditions at Layer 2 and higher look for protocol events.

In searching the data stream byte by byte, Layer 1 data conditions behave similarly to **Receiver** conditions on the Trigger Setup screens. This is another way of saying that the 16 trigger menus constitute a Layer 1 test. This test has a single state that is always current. Trigger menus with selections made on them are always active.

The three data conditions are DTE, DCE, and RECEIVE (for Port 1 when using a dual–port TIM). When one of these conditions is selected, a new rack of softkeys appears. The new options are shown in Figure 32-2.

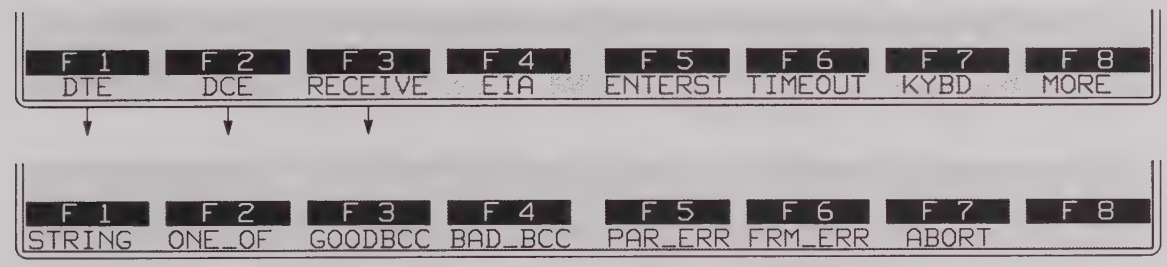


Figure 32-2 A spreadsheet trigger will monitor either data lead looking for these events.

(B) DTE

When DTE is selected, data on the TD lead will be monitored (on the main port when using a dual−port TIM).

(C) DCE

This condition monitors the RD lead (on the main port when using a dual−port TIM).

(D) Receive

This condition is intended for use in the emulate modes. It allows you to change the emulate mode of a program on the Line Setup screen without modifying the spreadsheet. When RECEIVE is selected, the INTERVIEW will always monitor the lead opposite its own transmit lead (on the main port when using a dual−port TIM). With **Mode:** EMULATE DTE as the Line Setup selection, the trigger will monitor RD. In Emulate DCE mode, the trigger will monitor TD.

(E) String

When a trigger monitors a data lead for a string, it searches for the exact, entire sequence of characters entered in the condition. Strings have a size limit of 32 characters. If constants are entered in the string, the 32−character limit is applied after all constants have been expanded.

After pressing the STRING softkey or typing STRING followed by a space, begin the string. Strings are always enclosed in quotation marks on the spreadsheet.

Here is an example of a Layer 1 data condition:

```
CONDITIONS: RECEIVE STRING "␍%R" WAIT_EOF
```

where WAIT_EOF delays trigger−true until the block of data holding the string has ended with a good block check.

(F) One−Of Character

When ONE_OF is entered, the trigger looks for any one of the characters in the list that follows. A single character in the data is all that is necessary to match a list. The effect of a "not−equal" character in a one−of list is explained in Section 25.3(I).

After pressing the ONE_OF softkey or typing ONE_OF followed by a space, begin the list. Lists and strings are always enclosed in quotation marks.

(G) Good or Bad BCC

BCC is partly a Layer 1 function, in that the calculation normally is a "hardware" function that tests the physical medium. It also is a Layer 2 function, in that the frame–check calculation is transmitted as part of the Layer 2 protocol. BCC, therefore, appears as a set of spreadsheet functions both at Layer 1 and Layer 2.

GOOD_BCC (good block–check calculation) and BAD_BCC can only be used as conditions when **Rcv Blk Chk** is turned on. **Rcv Blk Chk** is a menu field on the Line Setup menu: see Section 5.

> **NOTE:** **Rcv Blk Chk** is *on* automatically when Format: BOP is the Line Setup selection.

Press the softkey for GOODBCC or BAD_BCC when you want the trigger to take action on receipt of the BCC. The INTERVIEW does the block–check calculation that the user has defined on the BCC Parameters menu and compares it with the received block–check characters. See Table 10-1 and Table 10-2 for the block–check calculations done by the INTERVIEW.

(H) Parity Error

PARITY_ERROR looks for an error in relation to the **Parity** selection made on the Line Setup menu.

(I) Framing Error

FRAMING_ERROR applies to start–stop formats (ASYNC and ISOC) and detects framing errors in relation to the **Stop Bits** field on the Line Setup menu.

(J) Abort

When **Format:** BOP has been selected on the Line Setup menu, you can enter ABORT as a trigger condition. In $^{7}_{E}$ –framed protocols, seven consecutive 1–bits in midframe constitute an abort.

(K) Enter Receive String

Enter strings and lists inside quotation marks. A list is a series of characters that can be matched by a single data character. (A string must be matched by a data string.) A one–of condition is an example of a list. All ASCII–keyboard, control, and hexadecimal characters are legal in a receive string or list. Of the special–character keys, NOT EQUAL, DON'T CARE, FLAG, and SHIFT–FLAG are valid. SHIFT–FLAG displays the sync symbol S on the screen, and causes a search for the sync pattern.

[BIT MASK] is not valid. Bit masks are entered in receive strings by the keying sequence illustrated in Table 33-1.

Constants are also legal in any character position in a list or string. See Section 33, Strings, for an explanation of these string–search tools.

(L) Wait for End Of Frame

After the double–quotation mark is entered to close a string or list, the final Layer 1 condition appears under [F1] on the rack of softkeys. The condition is WAIT_EOF, or "wait for the end of the frame" before coming true and taking any actions. See Figure 32-3.

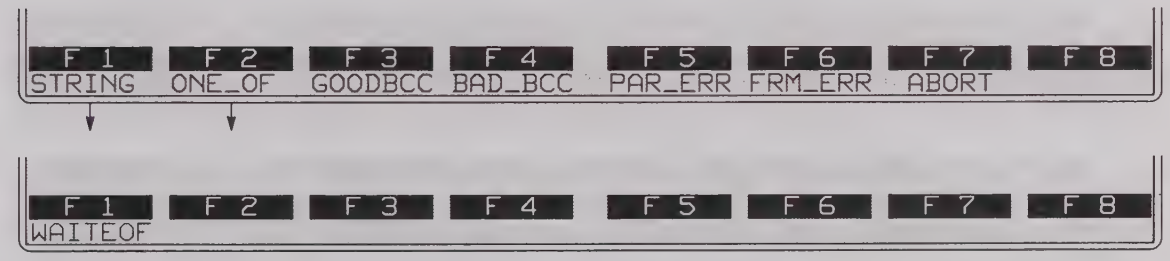


Figure 32-3 String and one–of conditions can be linked to a good BCC at the end of the frame ("EOF").

The WAIT_EOF condition does not occur above Layer 1, since data is not passed up to those layers until the frame is completed.

(M) EIA

Layer 1 conditions can monitor the status of six RS–232/V.24, V.35, or RS–449 control leads plus an additional seventh lead, the user–assigned (UA) input jack on the RS–232/V.24, V.35, or RS–449 test–interface module (TIM). Leads available for triggering are RTS, CTS, CD, DTR, DSR, and RI.

The EIA condition is a transitional/status condition. This means that when it is used alone it is true only if it transitions to true; but used in a trigger in combination with other conditions, it retains its status of on or off without having to transition to either status. The rules for combining conditions are explained in Section 31.2.

After pressing the EIA softkey or typing EIA followed by a space, make your lead selection from the upper rack of softkeys in Figure 32-4. Then select a status of ON or OFF.

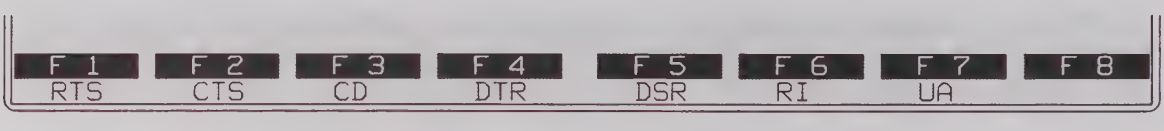


Figure 32-4 EIA leads monitored by the spreadsheet program.

For the standard RS−232/V.24 interface, ON implies that a lead is more positive than +3 volts with respect to signal ground. OFF implies only that a lead is not at or above the ON threshold, not necessarily that a minus threshold has been attained.

This is an example of an EIA condition:

```
CONDITIONS: EIA DTR OFF
```

(N) Xmit Complete

"SENDing" a transmission means queueing a transmission to send. The layer protocol (the RTS−CTS handshake, for example, at Layer 1) may delay the actual transmission. The XMIT_COMPLETE condition (selectable in the bottom rack of softkeys in Figure 32-1) will not come true until the transmission actually has been sent. Use this condition to start accurate response−time measurements.

32.2 Dual−Port Layer 1 Conditions

To bring up the bank of softkey conditions when using a dual−port TIM at Layer 1, first press the CONDITIONS softkey. This key becomes available when the cursor enters a programming block at the state level.

The first three condition softkeys—DTE, DCE, and RECEIVE—belong to Layer 1 for the main port. These are followed by generic conditions discussed in Section 31. The set of dual−port Layer 1 conditions is shown in Figure 32-1. The softkey for a fifth Layer 1 condition—XMIT_COMPLETE—appears on the second rack of condition softkeys shown at the bottom of the figure. The last three condition softkeys—DTE2, DCE2, and RECV2—belong to Layer 1 for the secondary port.

The Layer 1 conditions are transitional only. Refer to Section 31.2 for a discussion of how conditions may be combined.

All of the conditions except for the three secondary port condition softkeys are explained in Section 32.1; the secondary port softkeys are explained in the following subsections.

Figure 32-5 Dual–port Layer 1 conditions.

(A) Data

The first three and the last three trigger conditions at Layer 1 can monitor one of the two data leads for a specific data event; the first three are for the main port and the last three are for the secondary port. This event can be any of several characters, a string of characters, a good BCC following the character or string, an error revealed by a block or parity check, and so on.

Data conditions at Layer 1 monitor the entire data stream. Conditions in other layers also check the data leads, of course, but conditions at Layer 2 and higher look for protocol events.

In searching the data stream byte by byte, Layer 1 data conditions behave similarly to **Receiver** conditions on the Trigger Setup screens. This is another way of saying that the 16 trigger menus constitute a Layer 1 test. This test has a single state that is always current. Trigger menus with selections made on them are always active.

The three data conditions for the main port are DTE, DCE, and RECEIVE; these are explained in Section 32.1. The three data conditions for the secondary port are DTE2, DCE2, and RECV2. When one of these conditions is selected, a new rack of softkeys appears. The new options are shown in Figure 32-6. See Section 32.1 for information on these options.

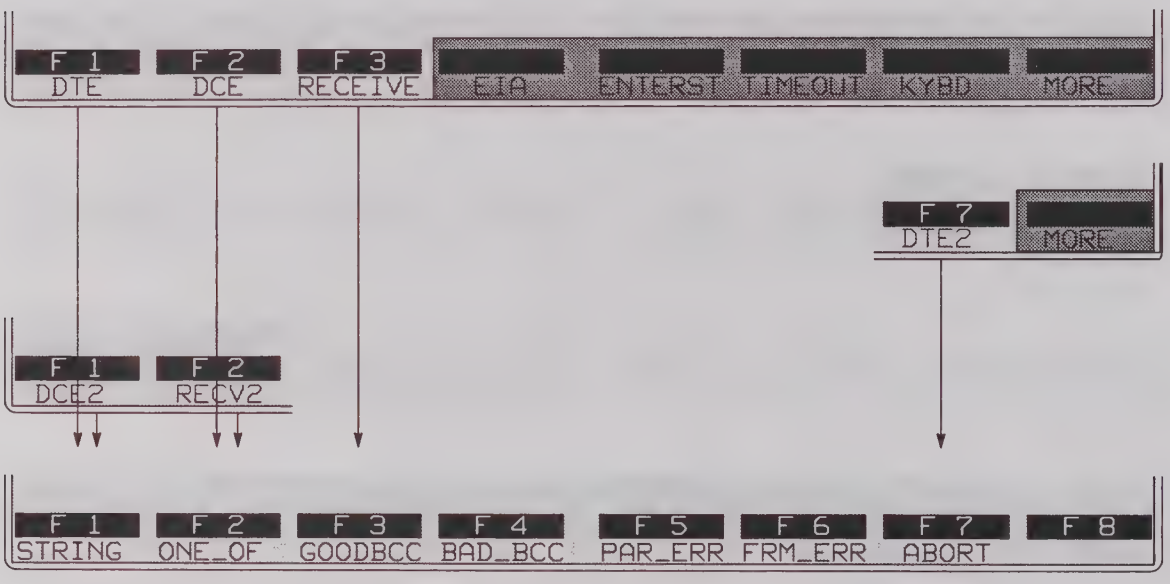


Figure 32-6 A spreadsheet trigger will monitor either data lead on the secondary port looking for these events.

(B) DTE2

When DTE2 is selected, data on the secondary port TD lead will be monitored.

(C) DCE2

This condition monitors the RD lead on the secondary port.

(D) Receive2

This Receive condition on the secondary port is intended for use in the emulate modes. It allows you to change the emulate mode of a program on the Line Setup screen without modifying the spreadsheet. When RECV2 is selected, the INTERVIEW will always monitor the lead opposite its own transmit lead on the secondary port. With **Mode:** EMULATE DTE as the Line Setup selection, the trigger will monitor RD on the secondary port. In Emulate DCE mode, the trigger will monitor TD on the secondary port.

32.3 Single–Port Layer 1 Actions

When a block of Layer 1 conditions has been entered, press DONE to access the softkeys for ACTIONS. The set of seven Layer 1 actions is shown in the softkeys in Figure 32-7. The names of these actions are SEND, EIA, OUT_SYNC, IDLE_LINE, ENHANCE, and CAPTURE. The other, darkened softkeys in the figure are layer–independent actions present at every layer, discussed in the previous section of this manual.

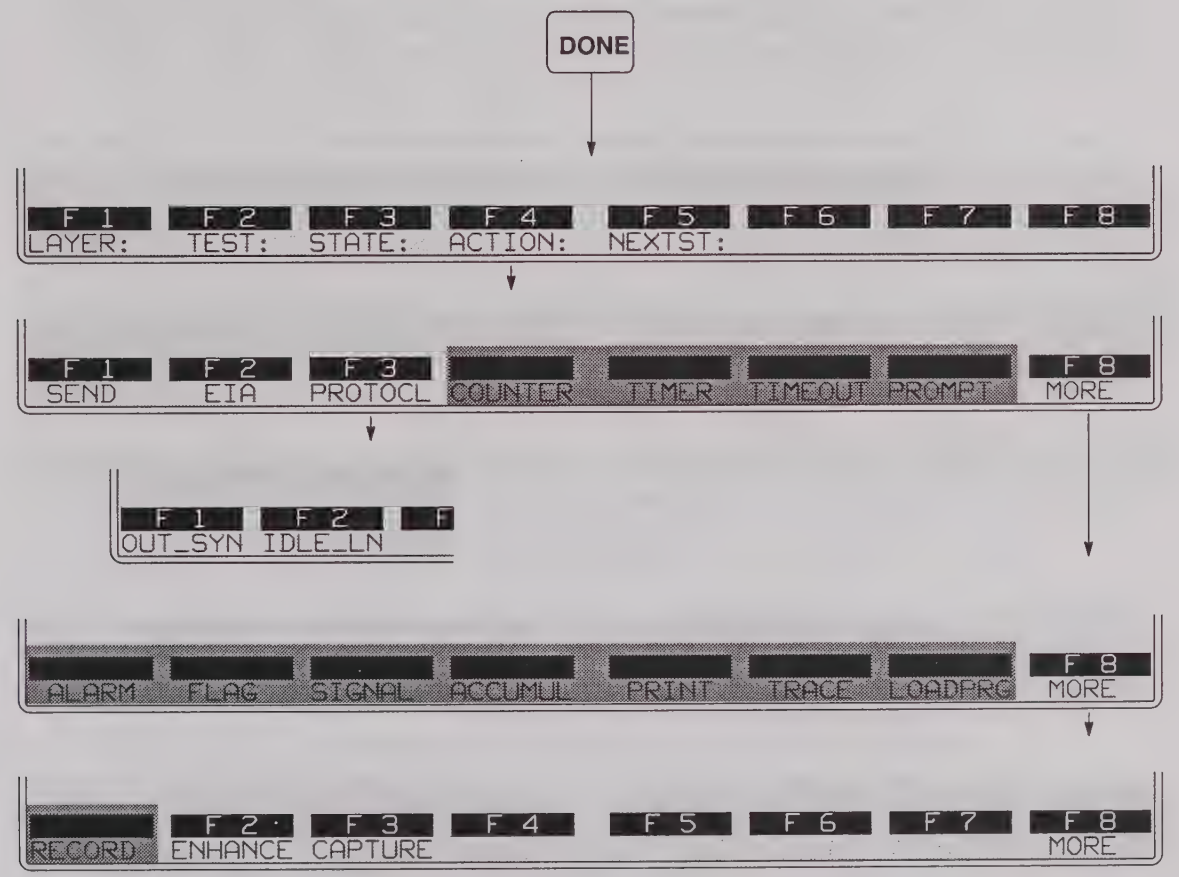


Figure 32-7 Layer 1 actions.

(A) Send

There is one SEND action—transmit a string—for single−port TIMs (or the main port on a dual−port TIM). While transmissions occur at all layers, only Layer 1 allows the user to type in a complete transmission, character by character. At higher layers, the user types the names of protocol elements and the software converts these mnemonics to strings. The user enters character strings directly at higher layers only into specified user−data fields.

The spreadsheet compiler identifies strings by the quotation marks surrounding them. Send−strings have no size limit (for practical purposes). All ASCII−keyboard, control, and hexadecimal characters are legal in a send−string. Special keys ([FLAG], [DON'T CARE], [BIT MASK]) are not legal.

To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)). Remember that the double parens are special characters produced by the CTRL–9 and CTRL–0 combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

Press the SEND softkey or type SEND followed by space to begin the entry. The prompt **Enter Transmit String** appears as in Figure 32-8. Enter the string inside quotation marks.

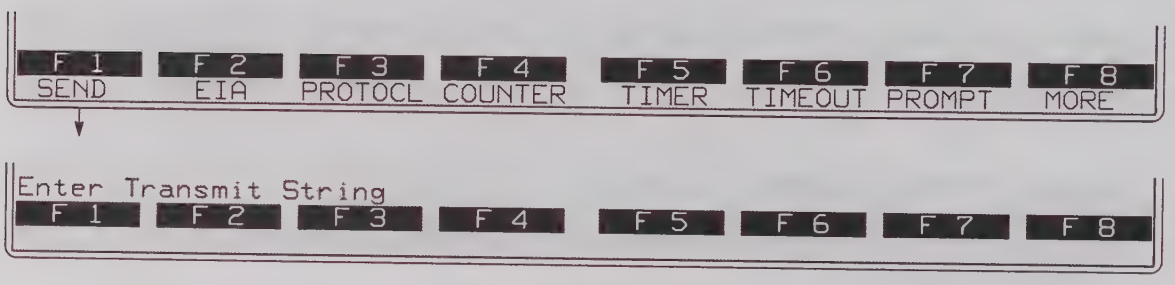


Figure 32-8 Transmit a string.

After quotation marks are typed in to close the transmit string, a set of softkeys appears for the error–checking value that will be appended to the transmit string. One of these must be selected; otherwise, the program will not compile and a **Premature End of File** error message is generated.

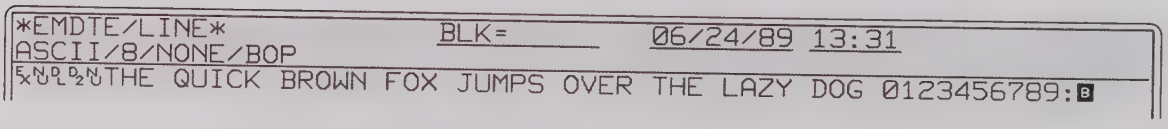


Figure 32-9 Select a block–check calculation to end the transmission.

1. *Good BCC.* This softkey entry allows you to append a good block–check sequence to your transmitted message. The INTERVIEW will make the proper calculation based on the parameters selected on the BCC Setup screen (see Section 10).

2. *Bad BCC.* Press the softkey labeled BAD_BCC to append an errorred block−check to your transmission. Assuming that **Rcv Blk Chk:** ON is the selection on the Line Setup menu, a BCC error will be indicated on the screen of the INTERVIEW by a B symbol. See Figure 32-10.

   For BOP format, the bad BCC will be CRC−16 instead of CCITT. For other formats it will be an inverted good BCC.

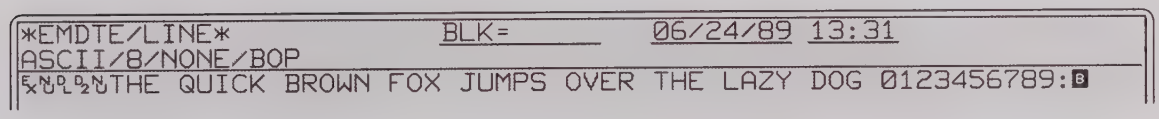


Figure 32-10 The INTERVIEW's TD monitor has detected a bad BCC transmitted by the unit's own TD driver.

3. *No BCC.* The NO_BCC softkey pertains to non−BOP formats only. Instead of appending a block−check calculation to a text message, the transmitter will revert directly to idle−line condition.

   Please note that receivers that are expecting BCC characters will treat the idle characters generated by the INTERVIEW as block−check characters. The INTERVIEW's own receivers (unless they go out of sync first) will display a bad−BCC symbol on the screen. (Refer to Figure 10-4.) The device under test probably will detect a BCC error and reject or ignore the message.

   The user may, of course, enter a good BCC "manually" as part of the text string that precedes the NO_BCC selection.

4. *Abort.* Abort is a BOP function only. Instead of appending a proper frame−check sequence (FCS), the transmitter will hold the lead at mark for eight bits (or longer if the transmitter is idling^{F_F}). Inside a frame, seven 1−bits in a row are sufficient to signal an abort.

   An aborted message is shown in Figure 32-11.

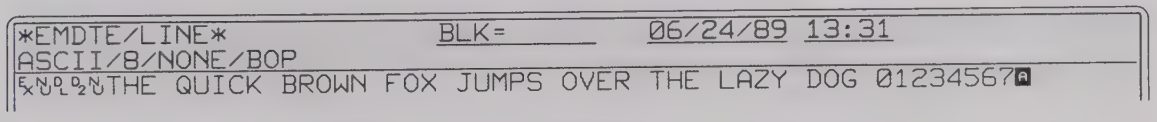


Figure 32-11 The INTERVIEW aborts a BOP frame by closing it with a byte of *FF* instead of *7E*.

(B) EIA

Press the softkey for EIA or type EIA followed by a space to bring five RS−232 leads and four auxiliary leads under spreadsheet control. The nine softkeys that represent EIA actions are illustrated in two separate racks of keys in Figure 32-12.

EIA actions are available only when the unit is in one of the emulate modes. A maximum of three RS−232 leads are controllable at one time. When **Mode:** EMULATE DCE is the Line Setup parameter, you control CTS, CD, and DSR. You may enter RTS ON or DTR ON as a spreadsheet action; but the DTE, not the INTERVIEW, controls these leads, and the actions will not take effect. To turn RTS or DTR on, first you must emulate a DTE.

The AUX softkeys allow you to apply off/on voltage to any of the AUX output jacks (four on the RS−232 Test Interface Module, three on the V.35 and RS−449 TIMs installed seated in the rear of the unit). (Refer to the documentation accompanying the interfaces.) These AUX outputs are useful for turning on and off a signal that is not a softkey selection or not under the control of your emulation.

> **NOTE:** The AUX actions on the spreadsheet have nothing to do with the 25−pin TTL AUXILIARY connector at the rear of the INTERVIEW.

After selecting a lead to control, select a status of OFF or ON. In the RS−232 specification for drivers, *on* is defined as +5 V to + 15 V while *off* means a range of −5 V to −15 V.

This is an example of an EIA action:

```
ACTIONS: EIA DTR ON
```

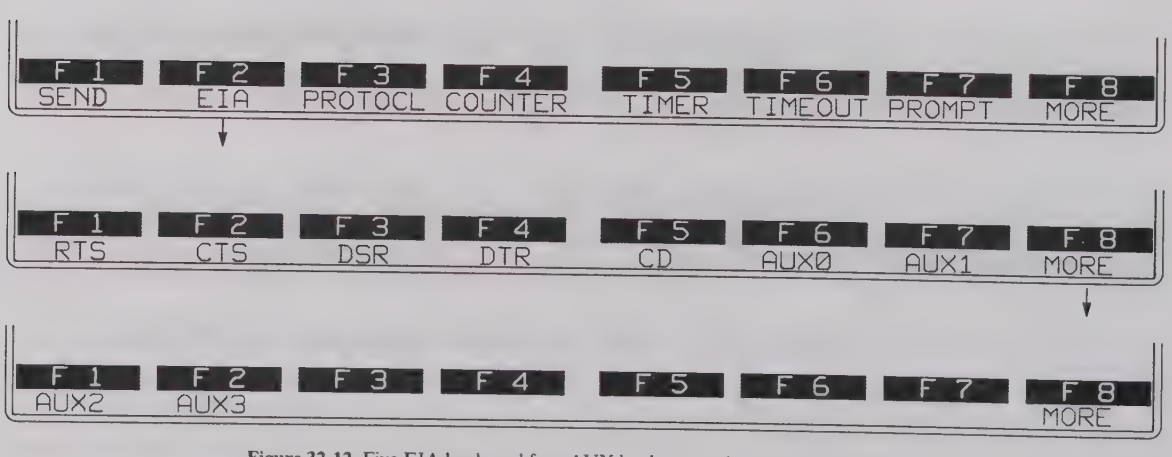


Figure 32-12 Five EIA leads and four AUX leads are under program control.

(C) Outsync

When the outsync action is taken, one or both receivers go out of synchronization from trigger true until the next synchronization pattern is received. All data that occurs in between outsync and resynchronization is considered "idle." If **Display Idle:** OFF is selected on the line setup, a receiver out of synchronization will prevent data from being presented to the screen and the character buffer as well as to the test program.

The outsync action also initiates the search for sync. Receivers that are already in sync do not look for sync. As soon as a receiver goes out of sync, the formatting logic begins to test for the one– or two–character sync pattern one bit at a time.

The outsync action may be useful when the information following a header group, for example, is of no interest. Simply go out of sync until the beginning of the next frame, when synchronization will restore the data display automatically. CAPTURE DTE (or DCE) OFF performs a similar function, except that "capture" must be turned on again by trigger when you want to resume the display.

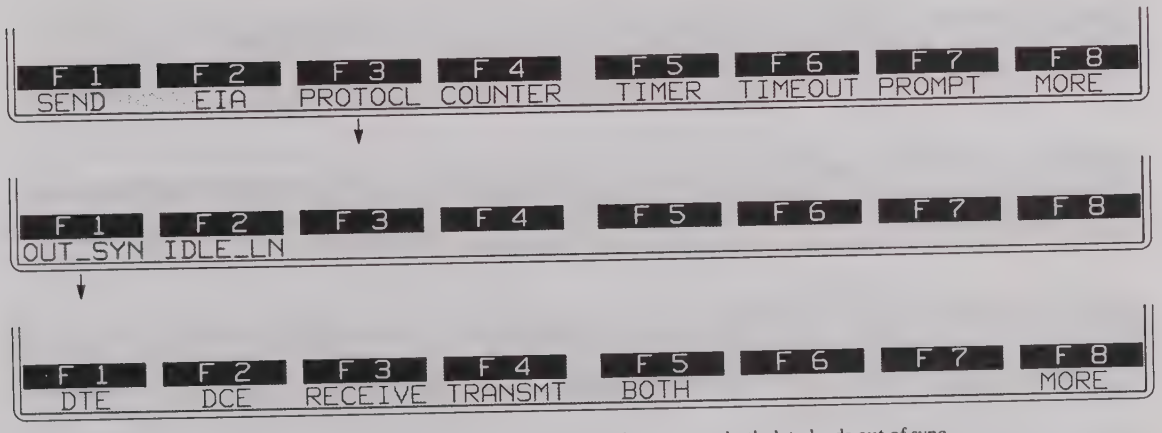


Figure 32-13 The spreadsheet program can force one or both data leads out of sync.

After you have pressed the OUT_SYN softkey or typed OUT_SYNC followed by a space, select one or both leads from the softkeys illustrated in Figure 32-13. RECEIVE and TRANSMT may refer to DTE or DCE, depending on your emulate mode at the moment. These selections allow you to change your emulation on the Line Setup menu without having to worry about changes to the spreadsheet program.

(D) Idle Line

IDLE_LINE allows you to use a trigger action to change the idle–line condition applied by the INTERVIEW. If you press the softkey for IDLE_LN (see Figure 32-13) or type IDLE_LINE followed by a space, the words **Enter Idle Character String** will appear on the prompt line in the softkey area at the bottom of the screen. Enter a single alphanumeric, control, or hexadecimal character in quotation marks. The red LED on the HEX key should be *on* for hexadecimal entry.

The idle–line action applies only when **Format:** SYNC has been selected on the Line Setup menu. This trigger action is useful for tests in protocols that employ different idle characters to signal changes in protocol state. An example is X.21 or X.21 *bis*, which in various states will idle ᶠF, ᴺ, +, and so on.

Here is an example of an IDLE_LINE action:

```
ACTIONS: IDLE_LINE "+"
```

(E) Enhance

The spreadsheet program can be used to enhance display data selectively. Data on either or both sides of the line may be enhanced. Figure 32-14 shows typical reverse–image enhancements. Enhancements are stored in the character buffer for later review: see Section 7.3.

Enhancements that pertain to the plasma display are reverse–image, blink, and hex. In addition to these, a low–intensity enhancement can be applied to data that is transmitted to a black–and–white monitor connected at the RS–170 port (see Figure 1-6).

Blink, reverse and low enhancements activated by the trigger–menu or spreadsheet program can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how blink, reverse, and low enhancements relate to character and background colors in the RGB output.

Enhancements are available at every protocol level, but only Layer 1 enhancements affect the raw–data display. Higher–level enhancements are applied to the protocol trace for a given layer.

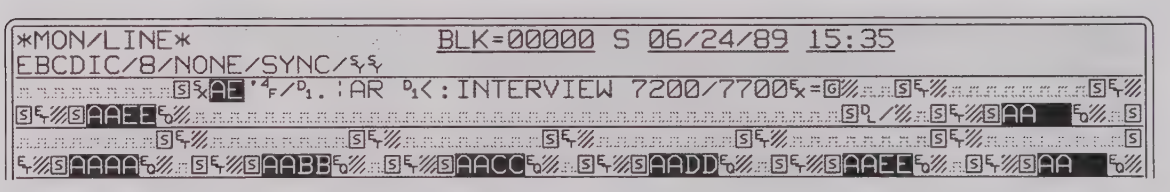


Figure 32-14 Enhancements may be used to highlight protocol fields.

After pressing the ENHANCE softkey or typing ENHANCE followed by a space, select one or both leads from the second level of softkeys in Figure 32-15.

Next, select the type of enhancement from the third tier of softkeys in Figure 32-15. Enhancements may be used in combination (such as reverse blink, or low–intensity reverse). Then at the final level, turn the enhancement ON or OFF.

1. *Reverse image*. Reverse–imaged characters are presented as dark letters on a lighter background.

2. *Low intensity.* This attribute does not affect data on the plasma display, which supports one display intensity only. Characters that are given this attribute will appear in low intensity on a CRT that is attached to the INTERVIEW through the RS−170 port.

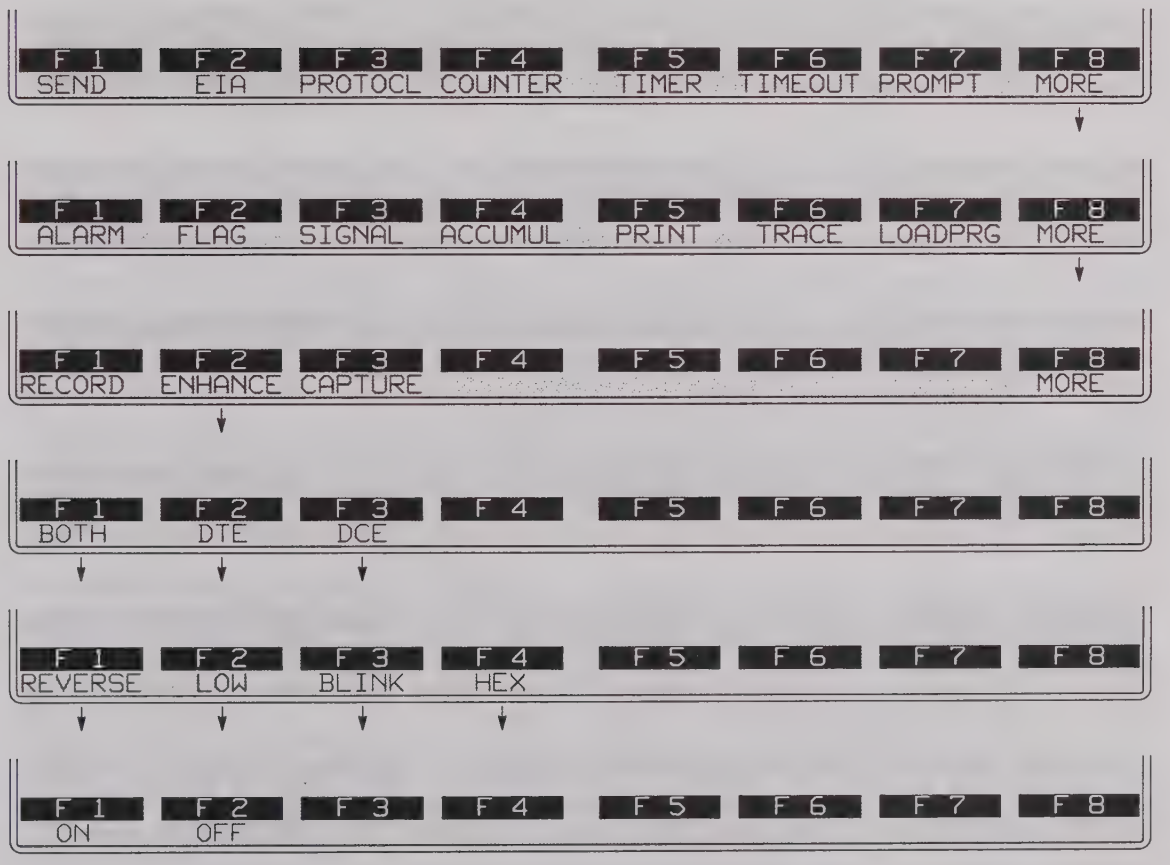


Figure 32-15 Layer 1 enhancements must be turned off as well as on by trigger.

3. *Blink.* BLINK causes data to be highlighted by a high−intensity area that blinks on and off. This is the most conspicuous highlight for small portions of data.

4. *Hexadecimal.* When the HEX enhancement is turned on, all data affected by the trigger is displayed in hexadecimal. Once data is stored in the buffer as hexadecimal, it remains in this format even if the [HEX] key is toggled.

   Refer to Figure 6-17 for data in which hex translation has been turned on for protocol characters and off for user (ASCII) data.

(F) Capture

This action turns on and off the presentation of data to the screen—that is, it stops or "freezes" the display—and capture of data to the screen buffer (character RAM). Unlike the Manual Freeze mode initiated by the FREEZE key, however, the "capture off" action does not allow you to scroll through the buffer while the test continues.

This action allows you to use the spreadsheet program to find important data and then preserve it in the buffer when it would otherwise be overwritten and lost.

Here is a sample capture action:

```
ACTIONS: CAPTURE BOTH OFF
```

where OFF means freeze the display and BOTH means with respect to DTE and DCE.

After pressing the CAPTURE softkey or typing CAPTURE followed by a space, select DCE, DTE, or BOTH from the rack of softkeys shown in Figure 32-16. On a subsequent set of softkeys, select ON or OFF as the capture action.

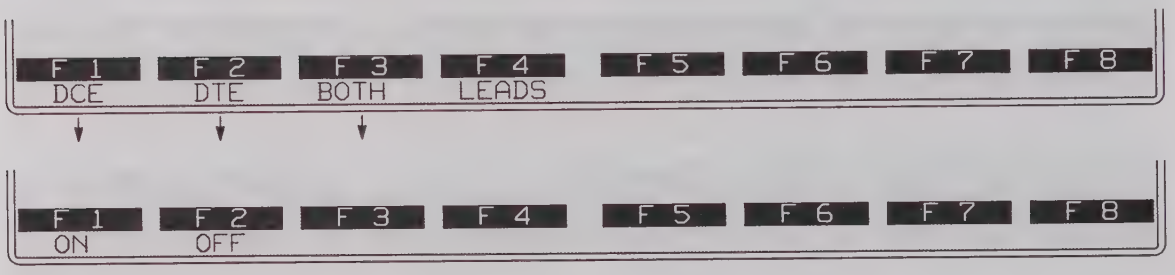


Figure 32-16 Screen display ("capture") can be turned on or off with respect to one data lead or both.

1. *DCE.* This option disables or enables the buffering and display of DCE (RD) data. Suppressing one data lead only does not serve the purpose of preserving data indefinitely in the buffer, since the other lead eventually will overwrite the buffer.

2. *DTE.* The TD lead by itself can likewise be suppressed or displayed.

3. *BOTH.* This option suppresses or displays all data.

4. *LEADS.* This option suppresses the lead changes to the display.

5. *ON.* This action enables buffering and display of the selected data.

6. *OFF.* This action suspends buffering and display.

32.4 Dual–Port Layer 1 Actions

When a block of Layer 1 conditions has been entered using a dual–port TIM, press DONE to access the softkeys for ACTIONS. The set of seven Layer 1 actions for dual–port use is shown in the softkeys in Figure 32-17. The names of these actions are SEND, OUT_SYNC, IDLE_LINE, ENHANCE, CAPTURE, and SEND2. With the exception of SEND2, they are all described in Section 32.3 of this manual. The other, darkened softkeys in the figure are layer–independent actions present at every layer, discussed in Section 31. SEND2, the secondary port send action, is described below.

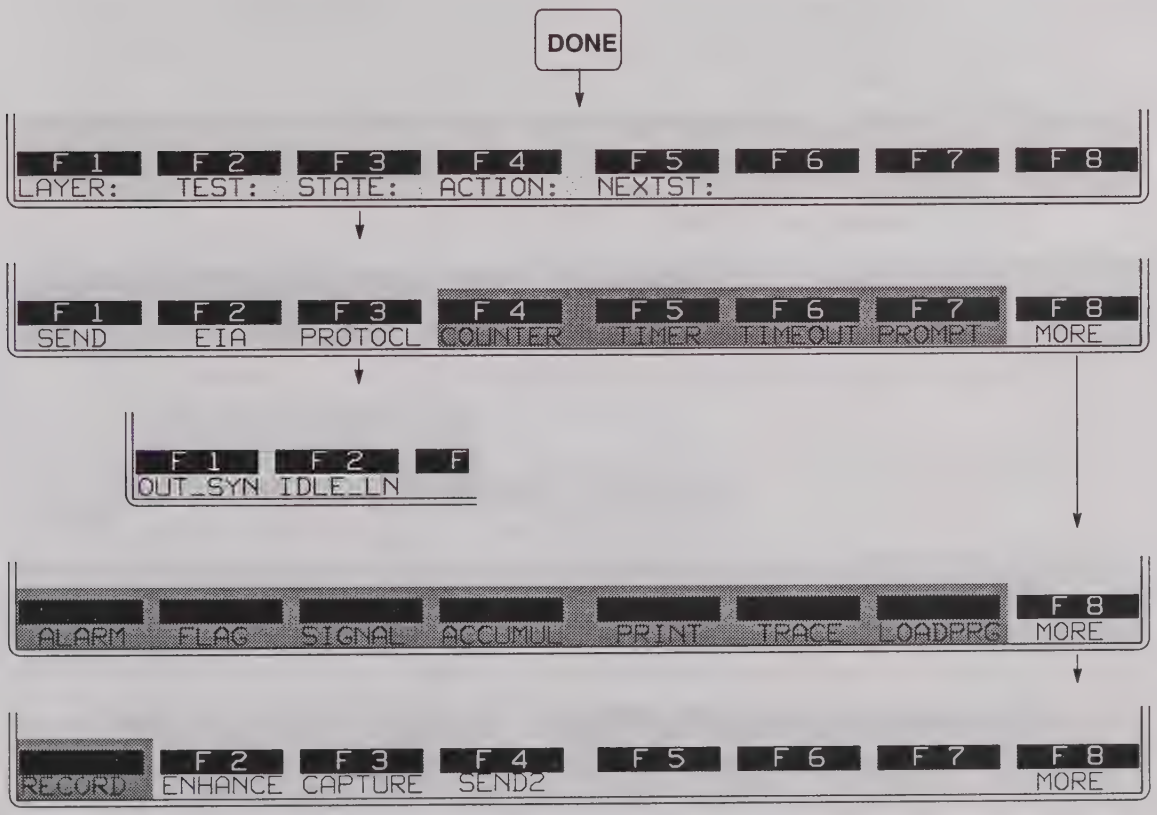


Figure 32-17 Dual–port Layer 1 actions.

(A) Send2

Transmitting a string is the only send action, and when using a dual–port TIM transmissions can be done on both the main port (using SEND) and the secondary port (using SEND2). While transmissions occur at all layers, only Layer 1 allows the user to type in a complete transmission, character by character. At higher layers, the user types the names of protocol elements and the software converts these mnemonics to strings. The user enters character strings directly at higher layers only into specified user–data fields.

Using a dual–port TIM, the SEND and SEND2 actions are executed in the same manner as the SEND action when using a single–port TIM.

The spreadsheet compiler identifies strings by the quotation marks surrounding them. Send–strings have no size limit (for practical purposes). All ASCII–keyboard, control, and hexadecimal characters are legal in a send–string. Special keys (FLAG, DON'T CARE, BIT MASK) are not legal.

To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)). Remember that the double parens are special characters produced by the CTRL–9 and CTRL–0 combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

For the main port, press the SEND softkey or type SEND followed by space to begin the entry; for the secondary port, press the SEND2 softkey or type SEND2 followed by space to begin the entry. The prompt **Enter Transmit String** appears as in Figure 32-8. Enter the string inside quotation marks.

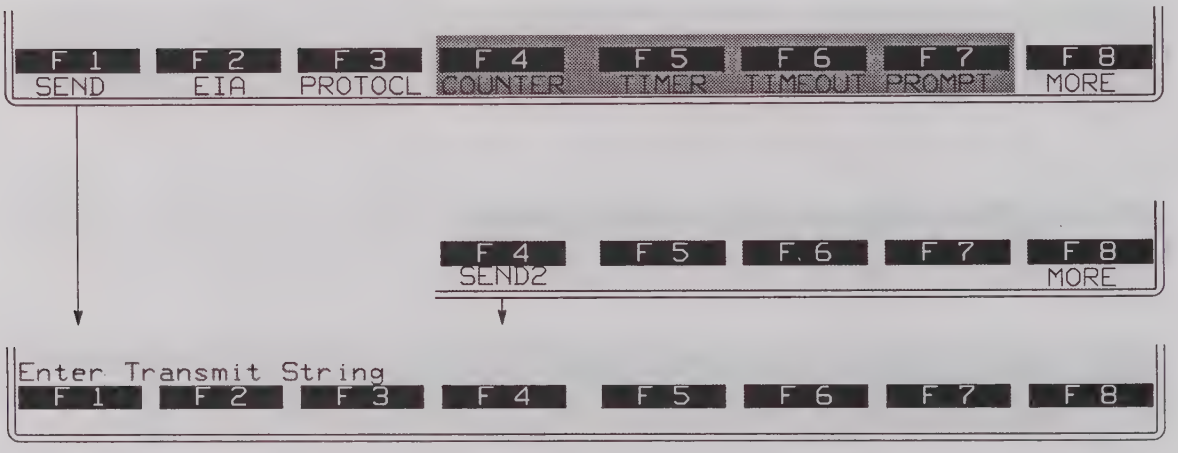


Figure 32-18 Transmit a string to the main port (SEND) and to the secondary port (SEND2) using a dual–port TIM.

After quotation marks are typed in to close the transmit string, a set of softkeys appears for the error–checking value that will be appended to the transmit string. One of these must be selected; otherwise, the program will not compile and a **Premature End of File** error message is generated.

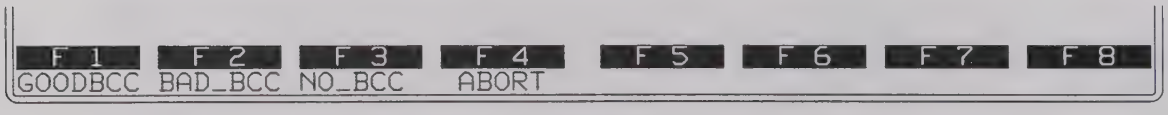


Figure 32-19 Select a block–check calculation to end the transmission.

See Section 32.3(A) for specific information on entering a block–check calculation to end the transmission as well as on the other softkeys.

33 Strings

A string on the Protocol Spreadsheet is a sequence of text characters that the operator encloses in quotation marks and enters following certain keywords. Strings are valid in both conditions (at Layer 1) and actions (at any layer). Depending on its use in the program, the string may be searched for, transmitted, printed out, or written to the screen while the program is running.

"Lists" are a subset of strings with an important distinguishing feature: where a string is a sequence of characters, a list is a set of single characters. Examples of lists are one–of conditions at Level 1 of the spreadsheet, or keyboard conditions at any level.

Apart from Layer 1 receive conditions and transmit actions at all layers (discussed below), strings are valid also in KEYBOARD conditions, where a list of keys may be entered, any one of which will satisfy the condition; in IDLE_LINE actions, where a single–character "string" entry represents the new idle character; in LOAD_PROGRAM actions, where the string must match the absolute pathname of the file to be loaded; and in PROMPT and TRACE actions.

All ASCII–keyboard, control, and hexadecimal characters are legal both in receive and transmit strings.

Two ASCII characters are treated in a special way. If you wish to include a quotation mark within a string, you must precede it with a backslash character (\"). If you wish to include a backslash character in a string, you must precede it with a second backslash character (\\). A single backslash is never included in the string.

Control characters are entered into text strings by the action of the CTRL key together with the key that bears the control–character mnemonic at the top right corner. Note that CR ("carriage return") is the mnemonic at the top right corner of the M key. Press CTRL–M to enter ↵ into a text string. The RETURN key does not produce a character entry.

Table 33-1
Valid Entries in Receive Strings

Type entry	Example	Key sequence	Example in string or list (1of)	This data satisfies string condition	Data beginning (arbitrarily) w/ AB satisfies 1of condition
ASCII	2	[2]	"123"	123	AB2
″	\″	[\] [SHIFT]-[']	"1\″3"	1″3	AB″
\	\\	[\] [\]	"1\\3"	1\3	AB\
Control	^V	[CTRL]-[V]	"1^V 3"	1^V 3	AB^V
Hex	^0B	[HEX] [0] [B]	"1^0B 3"	1^0B 3	AB^0B
Not Equal	≠	[NOT EQUAL] [2]	"1≠3"	113	A
Bit Mask	((XXXX1111))	[CTRL]-[9] [X] [X] [X] [X] [1] [1] [1] [1] [CTRL]-[0]	"1((XXXX1111)) 3"	1^0F 3	AB^0F
Not equal to bit mask	≠XXXX1111))	[NOT EQUAL] [CTRL]-[9] [X] [X] [X] [X] [1] [1] [1] [1] [CTRL]-[0]	"1≠XXXX1111)) 3"	123	A
Don't Care	[X]	[DON'T CARE]	"1[X]3"	153	A
Flag	[F]	[FLAG]	"1[F]3"	1[F]3	AB[F]
Sync	[S]	[SHIFT]-[FLAG]	"1[S]3"	1[S]3	AB[S]
Constant	((A)) where A is defined in a CONSTANT field as A = "abcdefg"	[CTRL]-[9] [SHIFT]-[A] [CTRL]-[0]	1((A)) 3	1abcdefg3	ABabcdefg

33.1 Strings To Be Matched Against Line Data

String conditions are legal in STRING and ONE_OF conditions at Layer 1 only.

Receive strings (and DTE/DCE strings) have a size limit of 32 characters. Their size cannot be expanded through the use of constants. (Any constants will be expanded *before* the size limit is enforced during compilation of the program.)

(A) Special Characters

Of the special–character keys, [NOT EQUAL], [DON'T CARE], [FLAG], and [SHIFT]–[FLAG] (for the [5] character) are valid. [BIT MASK] is not valid. Bit masks are entered in receive strings by the keying sequence illustrated in Table 33-1.

(B) Embedded Strings ("Constants")

The string represented by a constant may be embedded in a receive string or a list. A constant is a textual string that is represented by a symbolic name. This name is inserted into a string or list inside double parens. Double parens are special non–ASCII characters produced on the keyboard by [CTRL]–[9] and [CTRL]–[0].

The following is an example of a constant used in a spreadsheet condition:

```
CONDITIONS:
  RECEIVE STRING "((ADDR_A))⁵₀"
```

The data that satisfies this string will depend on the definition of the constant. Here is one possible definition:

```
CONSTANTS:
  ADDR_A = "AA⁷ꜰ⁷ꜰ"
```

The data that satisfies the condition will include the expanded constant along with the rest of the string: AA⁷ꜰ⁷ꜰ⁵₀.

33.2 Strings To Be Transmitted

Only Layer 1 allows the user to type in a *complete* transmit string, character by character. In the following transmit string, the entire transmission including sync characters, is inside quotation marks:

```
SEND "⁵ᵧ⁵ᵧ⁰ʟ⁷₀" NO_BCC
```

At higher layers, the user types the names of protocol units and values as "keywords" and the software converts these elements to strings. Immediately following the keyword entries, the user may add a string in quotation marks. Here is an example of a string following nonstring entries in Layer 2 SDLC:

```
SEND FRMR ADR=C1 P/F=1 "¹₁⁶₆⁰₈" GDBCC
```

All ASCII–keyboard, control, and hexadecimal characters are legal in a transmit string. None of the special–character keys (NOT EQUAL, BIT MASK, DON'T CARE, FLAG) is valid.

(A) Constant

Constants may be transmitted. Simply place the name of the constant inside double parens and insert the unit into the string. While the test is being compiled, the constant is replaced in the string by the text that is assigned to it.

The canned "fox" message is a built–in constant named FOX that is defined internally as follows: FOX = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 0123456789." An example of the FOX constant, as it appears in a transmit string, is given in Table 33-2.

(B) Transmit Variables

Certain variables may be transmitted also. Any number of counters and flags may have their values transmitted at any point.

If a counter or flag is named inside double parens in a transmit string, the current hexadecimal value of the *low–order byte* of that counter or flag is transmitted with the rest of the string. An example of a counter used in a transmit string is given in Table 33-2.

In order to be referenced in a transmit string, a counter or flag must first be created in a trigger–menu or spreadsheet condition or action. The counter or flag need not be named on a statistics screen.

Do not name a counter (or flag) in a transmit string if it has the same name as another flag (or counter). It is unpredictable which one will be transmitted.

The low–order byte of a counter or flag is the default byte to be transmitted. The second byte will be transmitted instead if the name of the counter or flag is followed by **[1]** inside the double parens. Here is an example of a Layer 2 transmission that includes both bytes of a flag named **seq** (as well as a fox message):

```
SEND INFO ADR=C1 NR=AUTO NS=AUTO "²C⁰0⁰2⁰1((seq[1]))((seq))⁰3⁹0⁸0 5F((FOX))¹3"
```

Flags are two bytes long, counters are four. All four bytes of a 32–bit counter may be transmitted. Here is a transmit string that sends a complete counter named **fourbyte**:

```
SEND " counter = ((fourbyte[3]))((fourbyte[2]))((fourbyte[1]))((fourbyte))"
```

(C) Data Request

A transmit string that is created at one protocol layer may be passed down transparently to lower layers, one layer at a time. A user–entered message that is sent down at Layer 4, for example, is detected at Layer 3 as an N_DATA REQ primitive and may be handed down to Layer 2 as an "((N_DATA))" string.

The string is appended either to a SEND DATA action (or to a DL_DATA REQ primitive). See the example below. The SEND DATA action will append a packet header to the N–data automatically. The DL_DATA REQ primitive will not add a header to the N–data string; but the user may enter additional data inside the quotation marks (*not* inside the double parens).

Layer 2, in turn, detects the data as a DL_DATA REQ primitive, and may hand it down to Layer 1 in the form of a "((DL_DATA))" string appended to a SEND INFO action (or to a PH_DATA REQ primitive).

```
LAYER: 4
      STATE: transport
         CONDITIONS: KEYBOARD " "
         ACTIONS: N_DATA REQ "((FOX))"
LAYER:3
      STATE: network
         CONDITIONS: N_DATA REQ
         ACTIONS: SEND DATA PATH= 0 "((N_DATA))"
LAYER:2
      STATE: datalink
         CONDITIONS: DL_CONNECT REQ
         ACTIONS: DL_CONNECT CONF
         CONDITIONS: DL_DATA REQ
         ACTIONS: SEND INFO "((DL_DATA))"
```

Data is sent up the layers also. The mechanism for passing data upward is the GIVE_DATA action included in the protocol personality package at each layer. Since the user will not normally wish to add protocol headers to upward–moving data, this data is not treated as a separable string inside quotation marks. It is passed upward transparently in the GIVE_DATA action.

Table 33-2
Valid Entries in Transmit Strings

Type entry	Example	Key sequence	Example in transmit string	Data transmitted
ASCII	2	[2]	"123"	123
″	\″	[\] [SHIFT]–[']	"1\″3"	1″3
\	\\	[\] [\]	"1\\3"	1\3
Control	$^{S}_{Y}$	[CTRL]-[V]	"1$^{S}_{Y}$3"	1$^{S}_{Y}$3
Hex	$^{0}_{B}$	[HEX] [Ø] [B]	"1$^{0}_{B}$3"	1$^{0}_{B}$3
Constant	((A)) where A is defined in a CONSTANTS field as A = "abcdefg"	[CTRL]-[9] [SHIFT]-[A] [CTRL]-[Ø]	"1((A))3"	1abcdefg3
Fox	((FOX))	[CTRL]-[9] [CAPS LOCK] [F] [O] [X] [CTRL]-[Ø]	"1((FOX))3"	1THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 01234567893
Counter or flag, (low–order byte)	((addr)) where addr is the name of a counter with a current decimal value of 14	[CTRL]-[9] [A] [D] [D] [R] [CTRL]-[Ø]	"1((addr))3"	1$^{0}_{E}$3
Counter or flag, second byte	((seq[1])) where seq is the name of a flag, the second (high–order) byte of which has a binary value of 01010100	[CTRL]-[9] [S] [E] [Q] [[] [1] []] [CTRL]-[Ø]	"1((seq[1]))3"	1$^{5}_{4}$3
Data in a data–request primitive	((DL_DATA)) at Layer 2, where Layer 3 string is fox message and Layer 3 header is $^{1}_{0}{}^{0}_{7}{}^{6}_{E}$	[CTRL]-[9] [D] [L] [SHIFT]–[-] [D] [A] [T] [A] [CTRL]-[Ø]	"1((DL_DATA))3"	$1^{1}_{0}{}^{0}_{7}{}^{6}_{E}$ THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 01234567893

34 OSI Primitives on the Protocol Spreadsheet

Figure 34-1 The softkey path for a *DL_CONNECT IND* condition primitive at Layer 3.

34 OSI Primitives on the Protocol Spreadsheet

Primitives are defined in the Open Systems Interconnect (OSI) Reference Model as protocol–independent interactions between adjacent layers of the model.

For example, data that comes into the INTERVIEW at Layer 1 or starts down the OSI "ladder" at a layer above Layer 1 is stored in a structure called an *IL* (interlayer) buffer. This buffer is passed between layers along with a primitive data unit (PDU), or primitive.

Since primitives are layer–specific, they are not available on the Trigger menus which offer conditions and actions at Layer 1. You must use the Protocol Spreadsheet to send, receive, or monitor primitives.

The Protocol Spreadsheet is divided into seven layers in accordance with the OSI model. By giving the operator control of the boundaries between these layers, primitives make layered programming possible.

Primitives for a given OSI layer may be entered in the Protocol Spreadsheet whether or not a protocol personality package is loaded in for that layer. Table 34-2 lists the primitives that may be entered on the current Protocol Spreadsheet. Due to the uncomplicated, "always–connected" nature of the RS–232/V.24, V.35, and RS–449 interfaces, Layer 1 primitives are automatic and do not appear on the Protocol Spreadsheet for that layer. OSI service for Layers 2 through 7 currently is available.

> **NOTE:** Unless a Layer 1 package (such as DDCMP) is loaded in, primitives are not available when **Format:** SYNC is selected on the Line Setup screen.

On the Protocol Spreadsheet, primitives take the form of conditions and actions. A condition primitive monitors the layer boundaries for action primitives that are sent down from above or up from below. An action primitive at any layer is sent either up or down to the next layer. Each primitive is shared by two layers. DL_CONNECT IND, for example, is an action primitive at Layer 2. At Layer 3, the same primitive is a condition. The prefix (DL) is an abbreviation for the name of the lowermost layer (Data Link) which shares the primitive. Table 34-1 lists all primitive prefixes and the layers which share them.

Table 34-1
Primitive Prefixes and Associated Layers

Prefix	Lowest Layer of Operation	Shares with Layer
PH	Physical (Layer 1)	2
DL	Data Link (Layer 2)	3
N	Network (Layer 3)	4
T	Transport (Layer 4)	5
S	Session (Layer 5)	6
P	Presentation (Layer 6)	7

34.1 Softkey Selections

The condition and action primitives specific to a given layer will be arrayed on softkeys that appear when you press the softkey for OSI. OSI is F7 on the second rack of condition softkeys. Figure 34-1 shows the softkey path to an OSI condition primitive at Layer 3. OSI is F1 on the third rack of action softkeys. Layer–specific softkey racks corresponding to the following general categories appear successively as selections are made:

(A) Direction

Indicate the direction from which the condition primitive will come. At Layer 3, for example, the first choice (DL) will detect primitives handed up from the layer below; the second selection (N) will detect primitives handed down from the layer above. As an action, you select the direction which you wish the primitive to go: the first choice (DL) sends the primitive to the layer below; the second selection (N) sends the primitive to the layer above.

(B) Type

Choose among the primitive types offered at each layer. Each layer has its own set of primitives, but they all can be grouped into four major phases: establishment, data transfer, release, and debug and error reporting.

In the establishment phase, a layer establishes a connection with the layer above and/or below. The activate and connect primitives provide this function. Data transfer is accomplished via the data, expedited data, and reset primitives. Deactivate and disconnect primitives break the connection between layers in the release phase. Debug and error reporting primitives include debug, error report, and unitdata.

(C) Request/Response

For some primitives, you must indicate whether you are searching for—or sending—a request (REQ or IND) or a response (RESP or CONF). INDications and CONFirms come from the layer below or go to the layer above; REQuests and RESPonses come from the layer above or go to the layer below.

(D) Path

Provide a path, if necessary. Interlayer primitives must handle channel or "path" information in order to insure that data moving down from Layer 4 is given the correct logical channel at Layer 3, or that data moving from Layer 3 to Layer 2 bears the correct frame address when it goes out on the data link.

A softkey sequence that leads to the PATH= selection for a primitive on the Protocol Spreadsheet is shown in Figure 34-2.

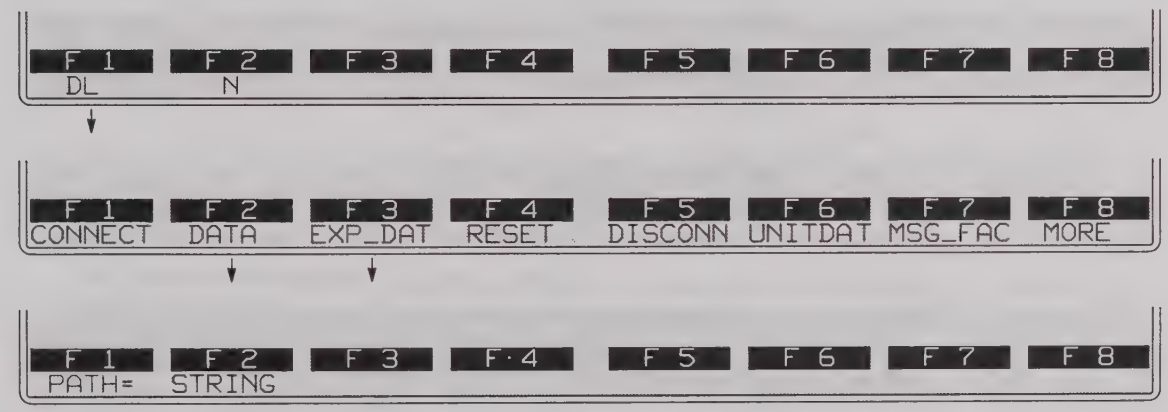


Figure 34-2 *DATA* and *EXPEDITE_DATA* action—primitives may carry path information as well as a data string.

Refer to Section 38.1(E) for a discussion of how paths are tied to call parameters (and directly or indirectly to logical channel numbers) via user entries on the Packet Level Setup screen (Figure 38-2) at X.25 Layer 3.

Primitive paths are only an important consideration when more than one layer is multiaddress or multichannel. In that situation, the vertical path numbers should match. Layer 3 might provide several logical channels, for example, while Layer 2 services more than one link address. When a set of call parameters is assigned by the user to path #1 at Layer 3, path #1 on the setup screen at Layer 2 should reference the appropriate link address for that call.

Remember that data primitives along with their path parameters usually are handled automatically (see Section 35). Automatic primitives will carry the same path information as the SEND or GIVE_DATA actions that generated them.

(E) String

Optional *strings* may be added to DATA or EXPEDITED_DATA action–primitives at any layer. A string is external data that is referenced in the list node of an interlayer buffer. (See Figure 62-1.) This buffer is passed with the selected primitive. One special use of the string field is to identify an IL buffer that has just been handed down from above. The macro N_DATA (or T_DATA or PH_DATA) enclosed in double parens in a data–primitive string field will identify the buffer that was just received from above. When the current action primitive is processed, the IL buffer will be passed to the layer below. One softkey sequence leading to a string selection is given in Figure 34-2. Always enclose a string in double quotation marks.

Here is an example of a data primitive at Layer 4 passing a string down to the next layer below:

```
LAYER: 4
      STATE: transport
         CONDITIONS: KEYBOARD " "
         ACTIONS: N_DATA REQ "((FOX)) "
LAYER:3
      STATE: network
         CONDITIONS: N_DATA REQ
         ACTIONS: SEND DATA PATH= 0 "((N_DATA)) "
LAYER:2
      STATE: datalink
         CONDITIONS: DL_CONNECT REQ
         ACTIONS: DL_CONNECT CONF
         CONDITIONS: DL_DATA REQ
         ACTIONS: SEND INFO "((DL_DATA)) "
```

This program is designed as a "quick" demonstration of OSI service primitives and will transmit a fox message out on the interface (and display it on the INTERVIEW screen) whether or not an actual link and call have been established. (Layer packages must of course be loaded in at Layers 2 and 3.) Note the following:

- The action at Layer 4 forces the fox message down to Layer 3.
- The SEND DATA action at Layer 3 adds an appropriate Layer 3 header to whatever data is referenced in the action.
- The string that contains the macro ((N_DATA)) indicates that the Layer 3 header should be copied into the IL buffer that was passed with the N_DATA primitive from Layer 4.
- The same SEND action at Layer 3 triggers an automatic DL_CONNECT REQUEST primitive, since Layer 3 does not send *packets* to Layer 2 unless the link has been established.
- The Layer 2 program bypasses link–establishment by forcing a DL_CONNECT CONFIRM primitive up to Layer 3.
- Now the data packet can be passed down to Layer 2, where a SEND INFO action inserts a frame header in the buffer received (in a DL_DATA REQ primitive) from Layer 3.

- Layer 1 primitives are automatic.

The fox message will also be transmitted if DATA REQUEST primitives are used instead of SEND actions at Layers 2 and 3 (or if no protocol packages are loaded); but the data in that case will not receive protocol headers.

34.2 Sample Primitives: CONNECT INDs and CONNECT REQs

Figure 34-3 and Figure 34-4 illustrate the flow of "connect" primitives between Layer 2 and Layer 3. The primitives in the figures are the labeled arrows positioned between the layers.

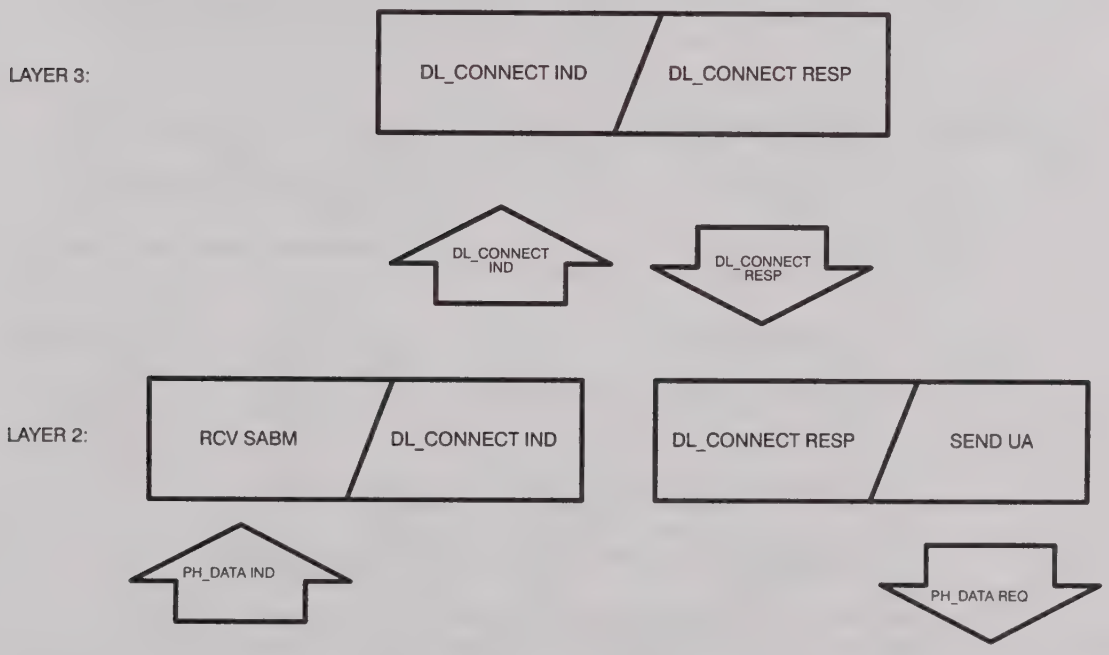


Figure 34-3 The (arrow–shaped) primitives moving between Layers 2 and 3 are intended to satisfy Layer 2 that a Layer 3 entity really is "up there." (The three rectangles contain spreadsheet conditions and actions.)

In Figure 34-3, Layer 2 receives a Set Mode command (SABM) from the data link. Before it responds positively (UA) to this command, Layer 2 passes up a DL_CONNECT IND primitive in order to verify that a Layer 3 entity really is "up there." When the active status of a Layer 3 entity is confirmed, Layer 2 sends the positive response (UA) down to Layer 1 and out onto the link to invite its Layer 2 peer to begin transferring data (Info frames).

The spreadsheet block that accomplished this exchange of primitives would be the following:

```
LAYER: 2
      STATE:establish_link
         CONDITIONS: RCV SABM
         ACTIONS: DL_CONNECT IND
         CONDITIONS: DL_CONNECT RESP
         ACTIONS: SEND UA
LAYER: 3
      STATE: dl_connect
         CONDITIONS: DL_CONNECT IND
         ACTIONS: DL_CONNECT RESP
```

In Figure 34-4, the request for confirmation of an adjacent layer is downward. Layer 3 wishes to send a Restart packet to the Layer 3 entity on the other side of the link; but it doesn't want to pass the packet down to Layer 2 if there is no mechanism at that layer to handle it. So Layer 3 precedes the Restart packet with a DL_CONNECT REQ primitive.

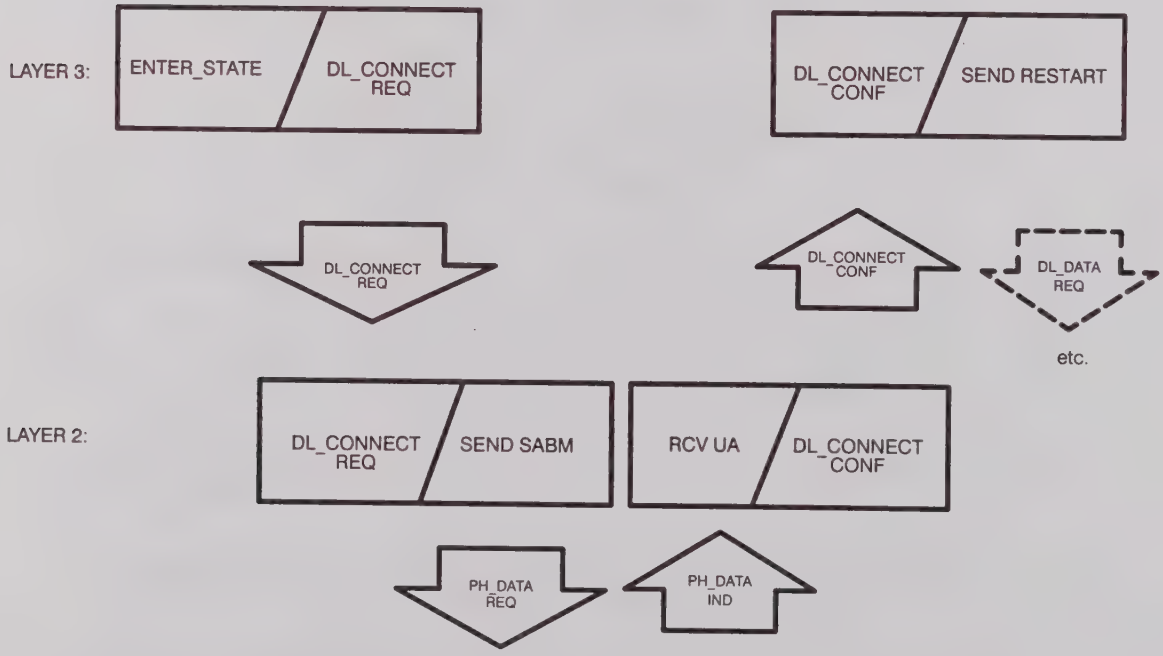


Figure 34-4 Layer 3 uses connect primitives to be sure that the Layer 2 entity below has established a link.

In the scenario illustrated in the figure, Layer 2 has not yet established the link. It responds to the connect–request primitive by sending a SABM, the X.25 command that initiates "connection" between link–level peers. When the SABM is acknowledged in a UA response, Layer 2 gives the DL_CONNECT CONF primitive up to Layer 3.

34.3 Sample Primitives: DATA INDs and DATA REQs

Figure 34-5 illustrates the primitives that are generated and monitored by the Protocol Spreadsheet when data is passed in both directions through an intermediate protocol layer (Layer 2). In this example, X.25 is the protocol package loaded in for both Layer 2 and Layer 3. Here a call–request packet is passed up through Layer 2 and received at Layer 3, and a call–confirm packet is sent down by Layer 3. The primitives in the figure are the labeled arrows positioned between the layers.

Data is in the form of physical–layer (PH) data when it moves in either direction between Layer 1 and Layer 2. PH_DATA primitives control the movement of this data. In between Layers 2 and 3, the data takes the form of data–link–layer (DL) data, with DL_DATA primitives responsible for data–delivery.

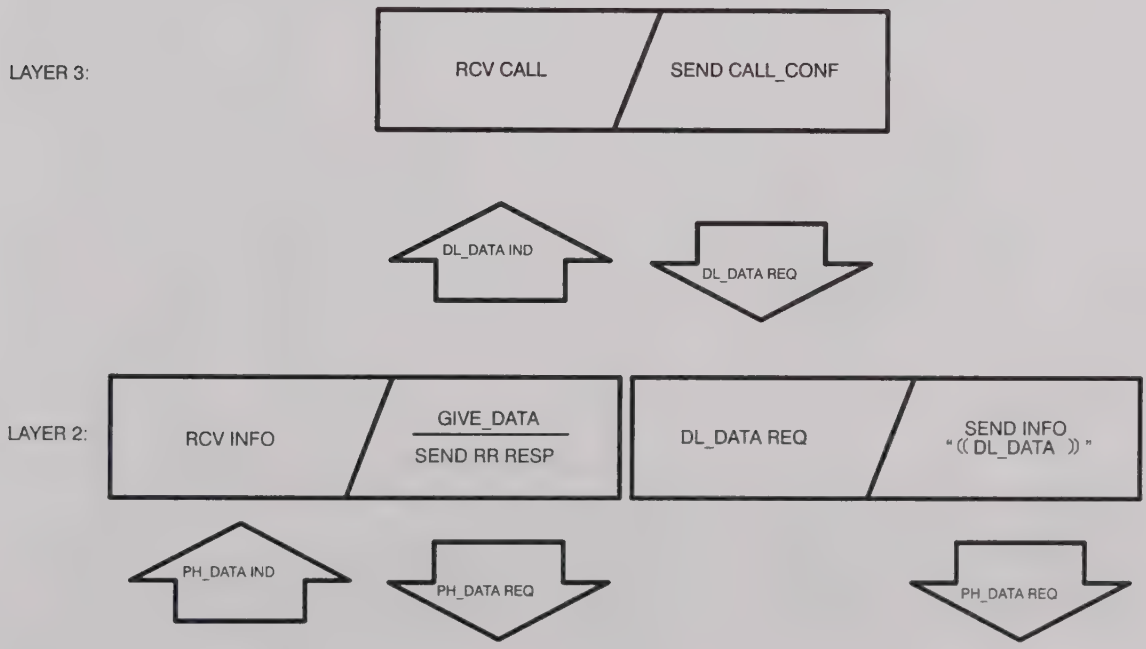


Figure 34-5 These (arrow–shaped) primitives are generated and monitored at Layer 2 when Layer 3 receives and sends data. (The three rectangles contain X.25 conditions and actions.)

The PH and DL versions of the data are not exactly the same. DL data has one less layer of protocol attached to it. In the example in Figure 34-5, the Layer 2 protocol was stripped off by the GIVE_DATA action when the call–request packet was being passed upward. On the call–confirm packet's trip down through the layers, Layer 2 protocol was added to the DL data by the SEND action—thus yielding PH data.

Note that "DL data" refers to data that moves above the DL layer (Layer 2), not below it. "DL data" can be taken literally to mean that as far as the DL layer is concerned, this is pure data, with no protocol that is recognizable at Layer 2.

When data is being passed upward, the primitives that signal the data are called indications (DATA INDs). When data is sent downward, the primitives at each layer are termed requests (DATA REQs).

Table 34-2
OSI Service Primitives

	Conditions		Actions	
Layer	From Layer Below	From Layer Above	To Layer Below	To Layer Above
7	P_CONNECT IND P_CONNECT CONF P_DATA IND P_EXPEDITED_DATA IND P_RELEASE IND P_RELEASE CONF P_UNITDATA IND P_ERROR_REPORT IND P_MGT_FACILITY IND P_DEBUG IND P_TD_DATA IND P_RD_DATA IND P_TD_EXPEDITED_DATA IND P_RD_EXPEDITED_DATA IND P_TD_UNITDATA IND P_RD_UNITDATA IND		P_CONNECT REQ P_CONNECT RESP P_DATA REQ P_EXPEDITED_DATA REQ P_RELEASE REQ P_RELEASE RESP P_UNITDATA REQ P_MGT_FACILITY REQ P_DEBUG REQ	
6	S_CONNECT IND S_CONNECT CONF S_DATA IND S_EXPEDITED_DATA IND S_RELEASE IND S_RELEASE CONF S_UNITDATA IND S_ERROR_REPORT IND S_MGT_FACILITY IND S_DEBUG IND S_TD_DATA IND S_RD_DATA IND S_TD_EXPEDITED_DATA IND S_RD_EXPEDITED_DATA IND S_TD_UNITDATA IND S_RD_UNITDATA IND	P_CONNECT REQ P_CONNECT RESP P_DATA REQ P_EXPEDITED_DATA REQ P_RELEASE REQ P_RELEASE RESP P_UNITDATA REQ P_MGT_FACILITY REQ P_DEBUG REQ	S_CONNECT REQ S_CONNECT RESP S_DATA REQ S_EXPEDITED_DATA REQ S_RELEASE REQ S_RELEASE RESP S_UNITDATA REQ S_MGT_FACILITY REQ S_DEBUG REQ	P_CONNECT IND P_CONNECT CONF P_DATA IND P_EXPEDITED_DATA IND P_RELEASE IND P_RELEASE CONF P_UNITDATA IND P_ERROR_REPORT IND P_MGT_FACILITY IND P_DEBUG IND P_TD_DATA IND P_RD_DATA IND P_TD_EXPEDITED_DATA IND P_RD_EXPEDITED_DATA IND P_TD_UNITDATA IND P_RD_UNITDATA IND

Table 34-2 (Continued)
OSI Service Primitives

	Conditions		Actions	
Layer	From Layer Below	From Layer Above	To Layer Below	To Layer Above
5	T_CONNECT IND	S_CONNECT REQ	T_CONNECT REQ	S_CONNECT IND
	T_CONNECT CONF	S_CONNECT RESP	T_CONNECT RESP	S_CONNECT CONF
	T_DATA IND	S_DATA REQ	T_DATA REQ	S_DATA IND
	T_EXPEDITED_DATA IND	S_EXPEDITED_DATA REQ	T_EXPEDITED_DATA REQ	S_EXPEDITED_DATA IND
	T_DISCONNECT IND		T_DISCONNECT REQ	
		S_RELEASE REQ		S_RELEASE IND
		S_RELEASE RESP		S_RELEASE CONF
	T_UNITDATA IND	S_UNITDATA REQ	T_UNITDATA REQ	S_UNITDATA IND
	T_ERROR_REPORT IND			S_ERROR_REPORT IND
	T_MGT_FACILITY IND	S_MGT_FACILITY REQ	T_MGT_FACILITY REQ	S_MGT_FACILITY IND
	T_DEBUG IND	S_DEBUG REQ	T_DEBUG REQ	S_DEBUG IND
	T_TD_DATA IND			S_TD_DATA IND
	T_RD_DATA IND			S_RD_DATA IND
	T_TD_EXPEDITED_DATA IND			S_TD_EXPEDITED_DATA IND
	T_RD_EXPEDITED_DATA IND			S_RD_EXPEDITED_DATA IND
	T_TD_UNITDATA IND			S_TD_UNITDATA IND
	T_RD_UNITDATA IND			S_RD_UNITDATA IND
4	N_CONNECT IND	T_CONNECT REQ	N_CONNECT REQ	T_CONNECT IND
	N_CONNECT CONF	T_CONNECT RESP	N_CONNECT RESP	T_CONNECT CONF
	N_DATA IND	T_DATA REQ	N_DATA REQ	T_DATA IND
	N_DATA_ACK IND		N_DATA_ACK_REQ	
	N_EXPEDITED_DATA IND	T_EXPEDITED_DATA REQ	N_EXPEDITED_DATA REQ	T_EXPEDITED_DATA IND
	N_RESET IND		N_RESET REQ	
	N_RESET CONF		N_RESET RESP	
	N_DISCONNECT IND	T_DISCONNECT REQ	N_DISCONNECT REQ	T_DISCONNECT IND
	N_UNITDATA IND	T_UNITDATA REQ	N_UNITDATA REQ	T_UNITDATA IND
	N_ERROR_REPORT IND			T_ERROR_REPORT IND
	N_MGT_FACILITY IND	T_MGT_FACILITY REQ	N_MGT_FACILITY REQ	T_MGT_FACILITY IND
	N_DEBUG IND	T_DEBUG REQ	N_DEBUG REQ	T_DEBUG IND
	N_TD_DATA IND			T_TD_DATA IND
	N_RD_DATA IND			T_RD_DATA IND
	N_TD_EXPEDITED_DATA IND			T_TD_EXPEDITED_DATA IND
	N_RD_EXPEDITED_DATA IND			T_RD_EXPEDITED_DATA IND
	N_TD_UNITDATA IND			T_TD_UNITDATA IND
	N_RD_UNITDATA IND			T_RD_UNITDATA IND

Table 34-2 (Continued)
OSI Service Primitives

	Conditions		Actions	
Layer	From Layer Below	From Layer Above	To Layer Below	To Layer Above
3	DL_CONNECT IND DL_CONNECT CONF DL_DATA IND DL_EXPEDITED_DATA IND DL_RESET IND DL_RESET CONF DL_DISCONNECT IND DL_UNITDATA IND DL_ERROR_REPORT IND DL_MGT_FACILITY IND DL_DEBUG IND DL_TD_DATA IND DL_RD_DATA IND DL_TD_EXPEDITED_DATA IND DL_RD_EXPEDITED_DATA IND DL_TD_UNITDATA IND DL_RD_UNITDATA IND	N_CONNECT REQ N_CONNECT RESP N_DATA REQ N_DATA_ACK_REQ N_EXPEDITED_DATA REQ N_RESET REQ N_RESET RESP N_DISCONNECT REQ N_UNITDATA REQ N_MGT_FACILITY REQ N_DEBUG REQ	DL_CONNECT REQ DL_CONNECT RESP DL_DATA REQ DL_EXPEDITED_DATA REQ DL_RESET REQ DL_RESET RESP DL_DISCONNECT REQ DL_UNITDATA REQ DL_MGT_FACILITY REQ DL_DEBUG REQ	N_CONNECT IND N_CONNECT CONF N_DATA IND N_DATA_ACK IND N_EXPEDITED_DATA IND N_RESET IND N_RESET CONF N_DISCONNECT IND N_UNITDATA IND N_ERROR_REPORT IND N_MGT_FACILITY IND N_DEBUG IND N_TD_DATA IND N_RD_DATA IND N_TD_EXPEDITED_DATA IND N_RD_EXPEDITED_DATA IND N_TD_UNITDATA IND N_RD_UNITDATA IND
2	PH_ACTIVATE IND PH_ACTIVATE CONF PH_DATA IND PH_RESET IND PH_RESET CONF PH_DEACTIVATE IND PH_ERROR_REPORT IND PH_MGT_FACILITY IND PH_DEBUG IND PH_TD_DATA IND PH_RD_DATA IND	DL_CONNECT REQ DL_CONNECT RESP DL_DATA REQ DL_EXPEDITED_DATA REQ DL_RESET REQ DL_RESET RESP DL_DISCONNECT REQ DL_UNITDATA REQ DL_MGT_FACILITY REQ DL_DEBUG REQ	PH_ACTIVATE REQ PH_ACTIVATE RESP PH_DATA REQ PH_RESET REQ PH_RESET RESP PH_DEACTIVATE REQ PH_MGT_FACILITY REQ PH_DEBUG REQ	DL_CONNECT IND DL_CONNECT CONF DL_DATA IND DL_EXPEDITED_DATA IND DL_RESET IND DL_RESET CONF DL_DISCONNECT IND DL_UNITDATA IND DL_ERROR_REPORT IND DL_MGT_FACILITY IND DL_DEBUG IND DL_TD_DATA IND DL_RD_DATA IND DL_TD_EXPEDITED_DATA IND DL_RD_EXPEDITED_DATA IND DL_TD_UNITDATA IND DL_RD_UNITDATA IND

35 Automatic OSI Primitives

Often the Protocol Spreadsheet primitives that operate below a given layer are handled automatically by the protocol package at that layer.

Data primitives are automatic any time a SEND or GIVE_DATA softkey action is entered on the spreadsheet.

Connect Requests are automatic when the first spreadsheet data primitive is passed downward. The DL_CONNECT REQ in Figure 34-4 does not have to be entered in the user program. The connect request (but not the confirm) is handled automatically by the layer–package software, which assumes that the user never wishes to pass data *downward* to an empty layer.

The Connect Ind and Connect Resp primitives in Figure 34-3, on the other hand, are not automatic. They are completely at the discretion of the programmer of the Protocol Spreadsheet. If the programmer wishes Layer 2 to complete the link setup and begin transferring Info frames without the active participation of a higher layer, that is a viable alternative.

In the sequence in Figure 34-5 all of the primitives designated by arrows—with one exception—are generated and monitored automatically by the RCV, GIVE_DATA, and SEND spreadsheet entries. (The lone exception is the DL_DATA REQ primitive that is used as a condition in Layer 2.) This automatic handling of primitives frees the user at the top layer from programming considerations outside of his own immediate protocol.

When protocol packages are loaded, monitor primitives (such as DL_TD_DATA IND) are passed up the layers automatically. These primitives allow the Layer 2 and Layer 3 protocol–trace screens to display frame and packet information even when emulate primitives have not been passed up.

Automatic handling of primitives will vary with different protocols. Refer to the sections on the individual protocols for information on which primitives are tied to which protocol conditions and actions.

36 X.21 Layer 1

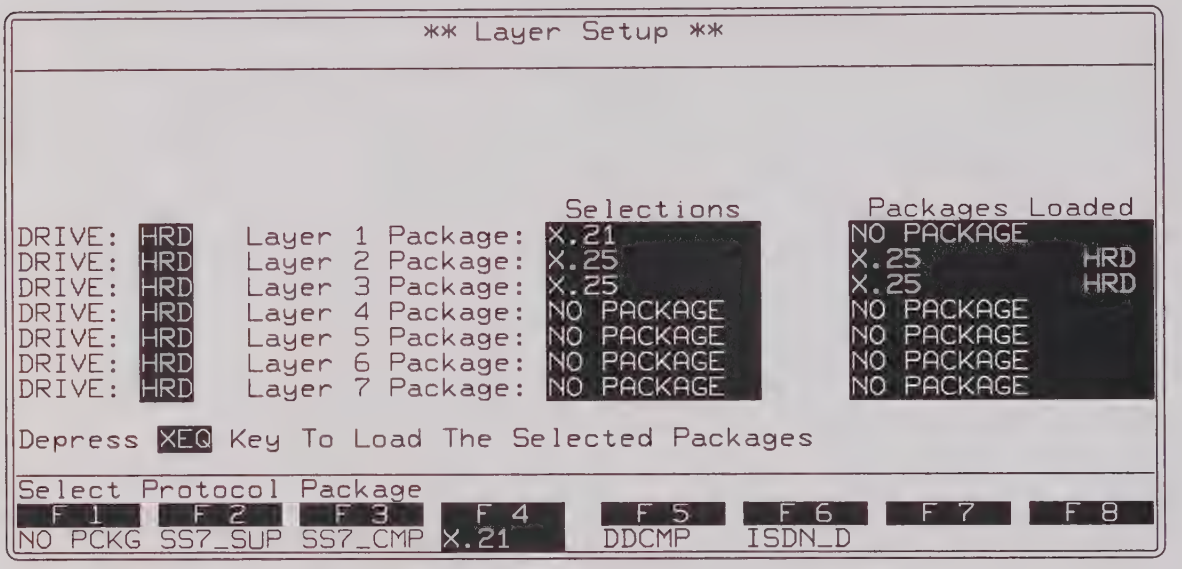


Figure 36-1 In addition to being an optional Test Interface Module, X.21 is a "layer–personality package" of softkey functions at Layer 1.

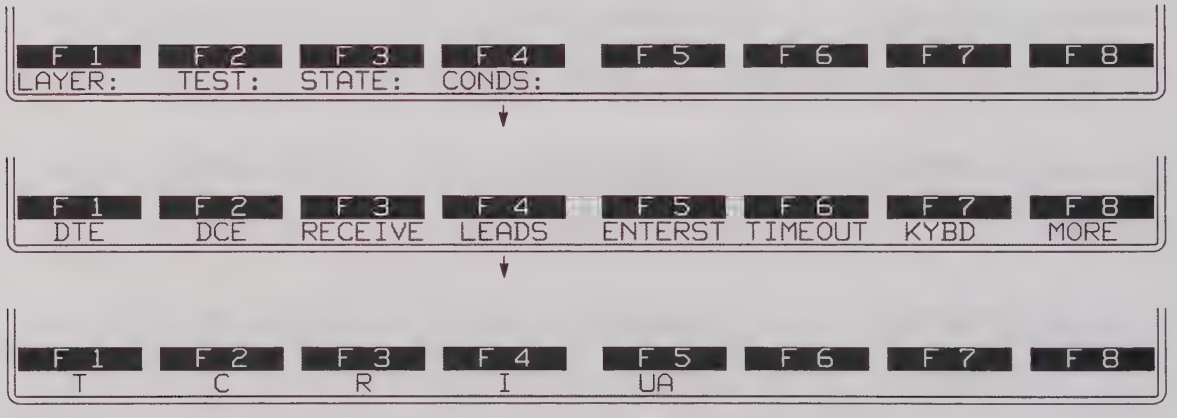


Figure 36-2 A special set of leads are available for monitoring once the X.21 package has been loaded in.

36 X.21 Layer 1

In addition to being an optional Test Interface Module (available as OPT−951−13−1), X.21 is a "layer personality package" of functions loaded into memory from disk via the Layer Setup screen. Figure 36-1 shows the Layer Setup screen configured to load in the X.21 package from the hard−disk drive. Refer to Section 8 for information on operating the Layer Setup screen.

The X.21 package consists of a group of conditions and actions at Layer 1 on the Protocol Spreadsheet that facilitate X.21 programming. Figure 36-2 shows the softkey path to a rack of condition softkeys that represent X.21 leads. These softkey conditions allow you to detect lead changes and lead status. Of the conditions on the first rack of softkeys below CONDS:, only LEADS is specific to X.21 and will be documented fully in this section.

Figure 36-3 shows the highest rack of softkeys containing actions that are specific to X.21. The SEND softkey includes a CALL_SETUP_SEND function that sends text messages always in ASCII code (consistent with X.21 call−setup protocol). The LEADS softkey gives the user control of X.21 control and data leads in emulation mode. The PROTOCL softkey includes functions that switch the line setup back and forth from ASCII 7−bit odd parity for call setups, to whatever line setup the user has configured for data transfer on the Line Setup menu.

Other softkey actions in Figure 36-3 are not specific to X.21 and are discussed elsewhere in the manual.

A group of Figures at the end of this section, Figure 36-10 through Figure 36-13, shows the INTERVIEW emulating either the user or the switch in calling, called, clearing, and cleared scenarios. The "conditions" and "actions" in these drawings are softkey conditions and actions in the X.21 Layer 1 personality package. The "new states" in the drawings are standard X.21 state names which may be borrowed as state names on the Protocol Spreadsheet.

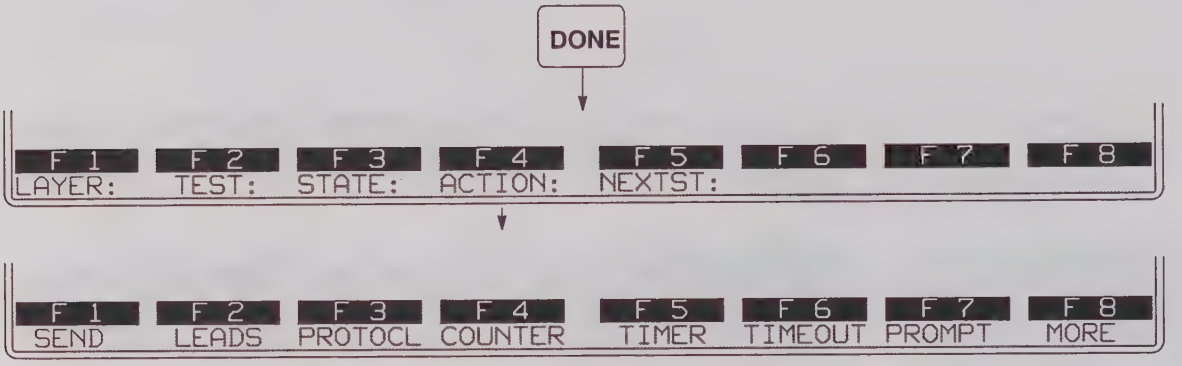


Figure 36-3 The *SEND*, *LEADS*, and *PROTOCL* softkeys branch to actions that are specific to X.21.

36.1 X.21 bis

The X.21 Layer 1 package also will work with the standard RS–232/V.24 TIM in an X.21 *bis* configuration. With the standard RS–232 TIM installed, the LEADS softkey shown in Figure 36-2 will be replaced by the EIA softkey that branches to standard EIA control–lead names: RTS, CTS, etc. The X.21 *bis* recommendation maps X.21 data and control leads to EIA leads according to the following conversions:

T	=	TD
C	=	DTR
R	=	RD
I	=	DSR

The LEADS softkey in Figure 36-3 changes to EIA in X.21 *bis*. When the RS–232 TIM is installed, data leads can be set to one of the standard X.21 idle conditions (+, ␁, ␖, ␕, and so forth) only via the IDLE_LN softkey action.

36.2 Transmitter/Receiver Phases

X.21 requires that data such as selection signals (the destination phone number) be transmitted during call setup. The data is transmitted in the following synchronous format: ASCII code, ␖␖ sync pattern, 7 data bits, odd parity, no BCC.

Once the call is established, a different format and code may be used at the link level and above. In order to monitor and transmit X.21 data and higher–level data correctly without exiting Run mode and reconfiguring the line setup, the X.21 layer package provides two different "phases" of the transmitter and the receivers. These phases are called CALL_SETUP and DATA_TRANSFER, and they are entered into the program via softkey. See Section 36.5(C), below.

When the program is in call–setup phase, data is monitored and sent according to the synchronous format and ASCII code defined above. In data–transfer phase, the format and code are as defined by the user on the Line Setup menu.

36.3 Sending From Layer 2

When Layer 1 is in data–transfer phase, a SEND action at Layer 2 will cause a transmission to go out onto the line automatically. No SEND action at Layer 1 is required.

When Layer 1 is in call–setup phase, a SEND action at Layer 2 will be ignored. If Layer 1 wishes to communicate to Layer 2 its readiness to send data, it must do so by SIGNAL action (see Section 31.4), since primitives are not currently operative at Layer 1.

36.4 X.21 Conditions

To bring up the bank of softkey conditions for Layer 1, first press the CONDITIONS softkey. This key becomes available when the cursor enters a programming block at the state level.

The first three condition softkeys—DTE, DCE, and RECEIVE—are common to all Layer 1 configurations. The fourth condition softkey, LEADS, is specific to the X.21 test–interface module. To the right of the LEADS softkey are general (layer–independent) conditions discussed in Section 31.

LEADS is a transitional/status condition and may be combined with other conditions (including other LEADS conditions). Refer to Section 31.2 for a discussion of how conditions may be combined.

(A) Data

The first three X.21 conditions can monitor one of the two data leads for a specific data event. This event can be any of several characters, a string of characters, a good BCC following the character or string, an error revealed by a block or parity check, and so on. When DTE is selected, data on the T lead will be monitored. A DCE condition monitors the R lead. RECEIVE conditions are intended for use in the emulate modes. When RECEIVE is selected, the INTERVIEW will always monitor the lead opposite its own transmit lead.

The fourth X.21 condition, LEADS, also can monitor both data leads. Figure 36-4 shows that T and R leads can be monitored for ZERO or ONE status. A data lead will satisfy one of these conditions when it is valid zero or valid one—that is, when it has retained its zero or one status for 16 consecutive bit times.

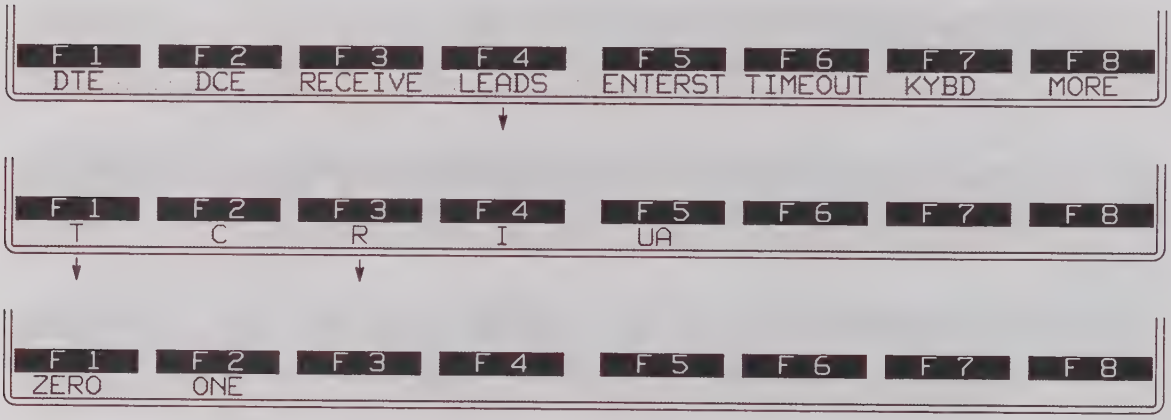


Figure 36-4 T and R leads can be monitored for *zero* or *one* status.

A mere transition from zero to one (or from one to zero) has no significance in X.21 protocol and cannot be detected by a LEADS T or LEADS R condition. (Control leads C and I, on the other hand, may be monitored either for a mere transition or for a valid status—see below.)

(B) Control Leads

X.21 conditions can monitor the status of C and I control leads. The C and I softkeys are in the conditions rack below LEADS. See Figure 36-5.

> **NOTE**: Before you may monitor the status of C and I leads, the buffering of control leads must be enabled on the Front–End Buffer Setup menu. See Section 9.1(B).

C and I may be tested for true status or for valid status. In the X.21 protocol, the state of the lead is valid if it has been true for 16 bit times. LEADS C ON, for example, checks the true state of the C lead. If the condition is alone in a CONDITIONS block, any momentary transition of the C lead from *off* to *on* will satisfy the condition. If the condition is used in a context where it is static rather than transitional—see Section 31.2 for a definition of this context—the true *on* state at the moment the lead is checked will satisfy the condition.

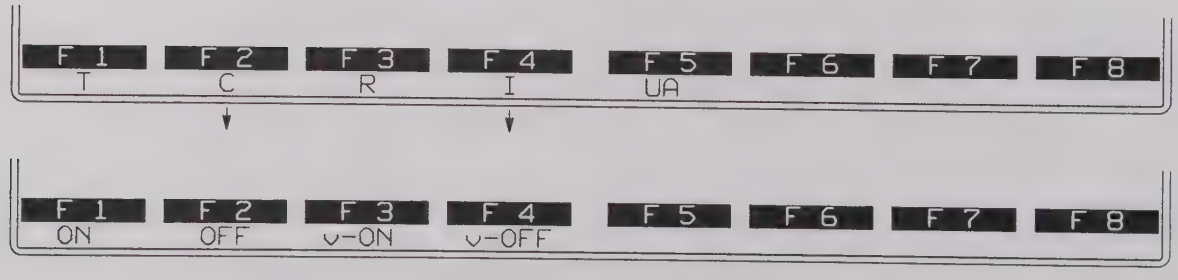


Figure 36-5 C and I leads may be tested for true status or for valid status.

The LEADS C ON_VALID condition requires not only that the state be true but also that it be valid. Valid conditions may be transitional or static, depending on how they are combined with other conditions in the same CONDITIONS block. When LEADS C ON_VALID is used alone in a CONDITIONS block, it is transitional. A transitional ON_VALID condition will be valid 16 bit times after the transition from off to on—assuming that it retains its true status for the entire 16 clock pulses.

(C) User Assigned

A LEADS UA condition detects an *on* or *off* state only if that state is valid. If a data lead is patched to the UA input, ON equals zero and OFF equals one.

36.5 X.21 Actions

When a block of conditions has been entered, press DONE to access the ACTIONS softkey. The actions that pertain to the X.21 Layer 1 personality package are SEND, LEADS, and PROTOCL, shown in Figure 36-3. SEND and LEADS actions are operative in emulate modes only.

(A) Data Leads

Data leads may be programmed to send character strings via the SEND softkey. They also may be programmed to idle constant mark, constant space, bell characters, plus characters, sync characters, and an alternating pattern of 0's and 1's via the LEADS softkey.

1. *Data–transfer send.* Figure 36-6 shows the three send options that branch under the SEND softkey. If you press SND_DTA, the keyword SEND is written to the spreadsheet screen and this prompt appears below the screen: "Enter Transmit String." This is a normal Layer 1 send action and it is appropriate whenever you are in Data Transfer state according to the X.21 protocol.

   Press the SND_DTA softkey or type SEND followed by space to begin the entry. Enter the string inside quotation marks. After quotation marks are typed to close the transmit string, a set of softkeys appears for the error–checking value that will be appended to the transmit string—GOODBCC, BAD_BCC, NO_BCC, or ABORT.

   To execute a data–transfer send, the program must be in data–transfer phase—see below. In this phase, the transmitter and receivers are obeying the code and format options selected on the Line Setup menu.

2. *Call–setup send.* The SEND softkey includes a CALL_SETUP_SEND function that sends text messages always in a code and format that is consistent with X.21 call–setup protocol. Press the SND_CLL softkey (Figure 36-6) or type CALL_SETUP_SEND followed by space to begin the entry. Enter the string inside quotation marks.

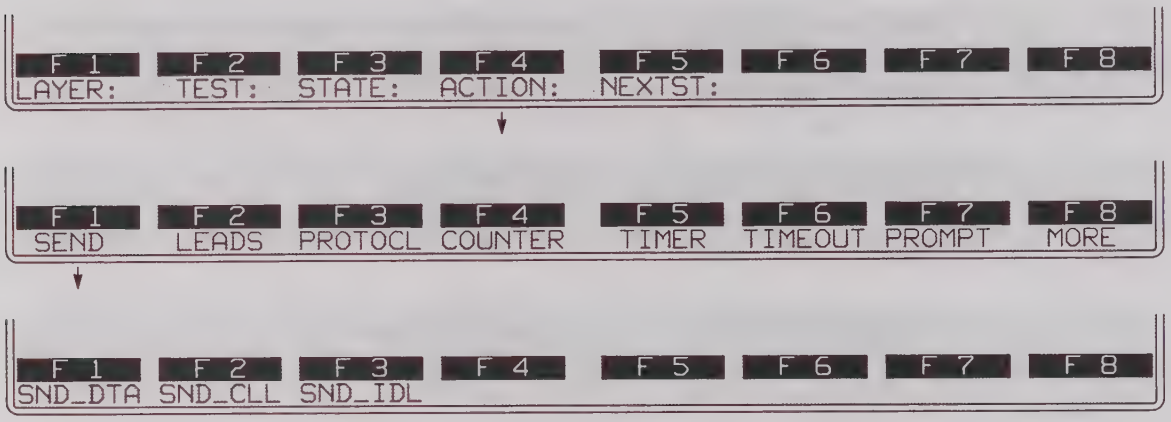


Figure 36-6 Two separate *SEND* actions are used to transmit either in data–transfer or call–setup format.

The code and format of a call–setup send action always is the same: ASCII code, 7 data bits, odd parity, no block check transmitted. Synchronization characters are ␖␖ (hex 1616). The synchronization pattern must be provided in the transmit string—it is not automatic.

To execute a call–setup send, the program must be in call–setup phase—see 36.5(C), below. In this phase, the transmitter and receivers are disregarding the code and format options selected on the Line Setup menu.

3. *Call–setup send idle.* The SEND softkey also includes a CALL_SETUP_SEND_IDLE function. This action combines the CALL_SETUP_SEND action and the LEADS action that specifies an idle character. When entered as these two separate actions, the change in idle may occur slightly before or after the transmission.

   Especially during high–speed operation, use the CALL_SETUP_SEND_IDLE action (and the NEW_IDL selection below it) to guarantee that the specified change in the idle character occurs *during* the string transmission. Press NEW_IDL and enter the (ASCII) idle character inside double quotation marks. Use the HEX key to enter hexadecimal characters.

4. *Idle.* Data leads also may be programmed to idle constant mark, constant space, bell characters, plus characters, sync characters, and alternating 0 and 1. Figure 36-7 shows the softkey path going through LEADS and T or R to the various idle states.

5. *One or zero.* Select ONE to idle constant mark, ZERO to idle constant space. Assuming that the program is in call–setup phase, a data lead idling mark will appear on the data display as ␏. Space idle will be displayed as ␑.

6. *Plus, bell, or sync.* Plus (+) characters, bell (␇) characters, and sync (␖) characters also may be transmitted as contiguous idle characters. A transmit

stream of any of these characters will be preceded by two ASCII sync characters (hex 16). In other respects, the action LEADS R BELLS has the same effect as the action IDLE_LINE "BL".

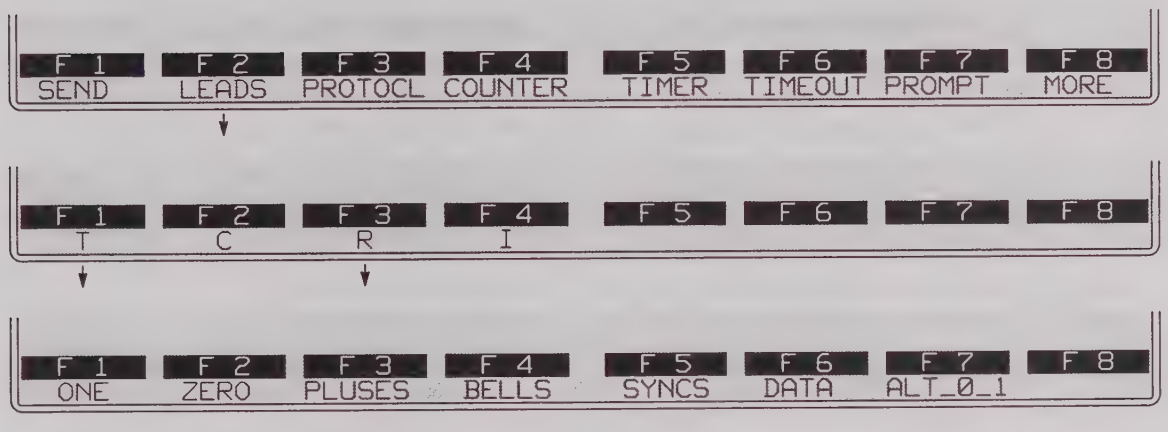


Figure 36-7 X.21 data leads T and R sometimes perform a "control" function by idling various characters.

Plus−, bell−, and sync−character idle actions do not take effect unless the program is in call−setup phase at the time the action is executed. The LEADS R ZERO action or the LEADS T ONE action will take effect even in data−transfer phase. The monitors may not be set up properly to detect or display the idle state, however, and we recommend that the programmer switch to call−setup phase as soon as one of the control leads first signals a clear request or indication.

7. *Data.* A ONE or ZERO leads action will clamp the line to the requisite voltage level. Once a lead is clamped, it must be unclamped before it can be used again for data. The DATA softkey shown in Figure 36-7 represents the "unclamp" action.

   To change from idling space to transmitting selection signals, for example, you would insert the unclamp action (LEADS T DATA) shown here:

```
STATE: call_request
   CONDITIONS: ENTER_STATE
   ACTIONS: LEADS T ZERO
      PROMPT "Press S to send selection signals"
   CONDITIONS: KEYBOARD "Ss"
   ACTIONS: LEADS T DATA
      CALL_SETUP_SEND "SYSY123123+"
```

   The PLUSES, BELLS, SYNCS, or ALT_0_1 action also will unclamp the line automatically.

8. *Alternating 0/1.* Press the softkey for ALT_0_1 to send an alternating series of zeroes and ones. The sequence is not preceded by sync characters. This idle action does not take effect unless the program is in call−setup phase at the time the action is executed.

(B) Control Leads

Control leads C and I may be controlled by spreadsheet action. Press the LEADS action softkey to bring up the rack of X.21 leads shown in Figure 36-8. Press C or I to access the softkeys that allow you to set a control lead to ON or OFF voltage.

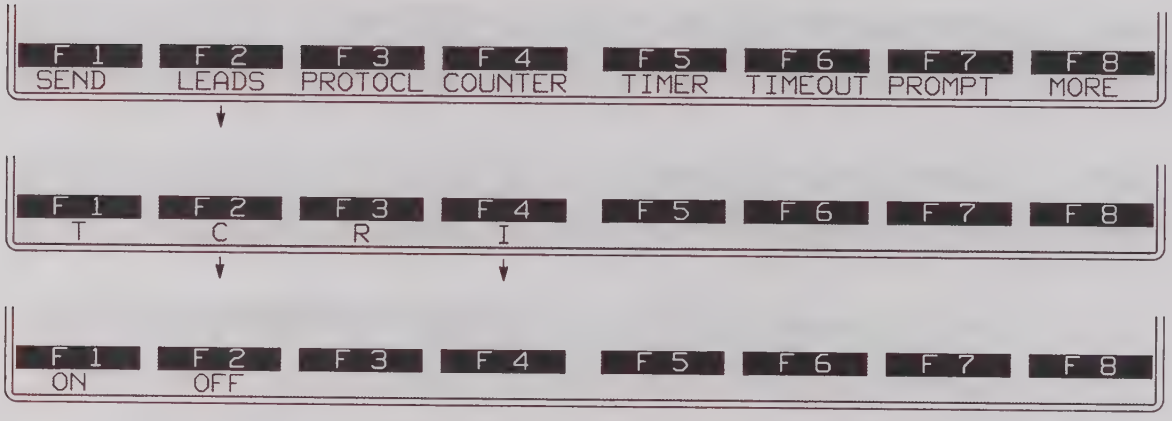


Figure 36-8 Control leads C and I may be turned on or off via softkey.

(C)Two Phases

The X.21 layer package provides two different "phases" of the transmitter and the receivers. These phases are called CALL_SETUP and DATA_TRANSFER, and they are entered into the program via softkey in the ACTIONS softkey rack that branches below PROTOCL. See Figure 36-9.

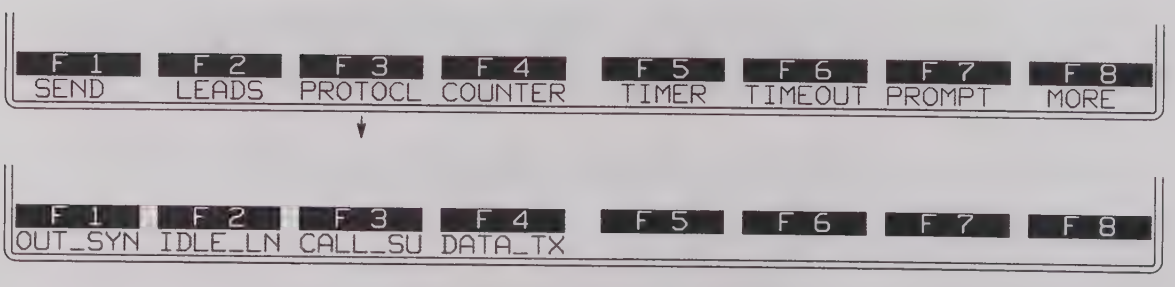


Figure 36-9 The X.21 layer package provides two different "phases" of the transmitter and the receivers, *Call Setup* and *Data Transfer*.

The initial configuration phase that the program adopts upon entering Run mode is selectable on the X.21 Interface Control setup menu. See the documentation that accompanies OPT−951−13−1. The default program−initiating phase is data transfer.

1. *Call setup.* When the program is in call−setup phase, data is monitored and sent according to the synchronous format and ASCII code defined above in Section 36.2. Idle display is automatically on. This means that when receivers encounter a condition that normally would send them out of sync (such as one or more $^{F}_{F}$ −idle characters), the receivers begin looking for sync as they normally would but the raw data continues to be displayed, in reverse video.

   With idle display automatically on, the transition will appear on the screen as a series of ␀ (NULL) characters when the data lead goes to zero to signal a call request or a clear request. For this reason, we recommend that the program adopt CALL_SETUP phase as soon as possible after a clear request or clear indication is signalled. In this way, the screen display of ␀ characters will record the clear request. In data−transfer phase, the steady zero signal will not be preceded by a special sync pattern and, depending on the line setup, may not be displayed.

   Figure 36-10 shows the INTERVIEW on the user side of the X.21 interface. Here the INTERVIEW adopts call−setup phase prior to clamping its leads to signal **dte_clear_request.**

   Figure 36-11 shows the INTERVIEW on the network−switch side of the X.21 interface. When the user side clears a call, the INTERVIEW programs a change to call−setup phase prior to clamping its leads to signal **dce_clear_confirmation.**

   The appropriate SEND action in call−setup phase is CALL_SETUP_SEND. See Section 36.5(A)2. The simple SEND action, appropriate for data−transfer phase, will not be executed in call−setup phase.

   When Layer 1 is in call−setup phase, a SEND action at Layer 2 will be ignored. If Layer 1 attains data−transfer phase and wishes to notify Layer 2 that it now is ready to send data, it must do so by SIGNAL action (see Section 31.4), since primitives are not currently operative at Layer 1.

2. *Data transfer.* When you press the softkey for DATA_TX (Figure 36-9) or type DATA_TRANSFER in an Actions block on the spreadsheet, you are sending the unit into data−transfer phase. In this phase, the unit monitors and sends according to the parameters selected by the user on the Line Setup menu. See Section 5.

   The appropriate SEND action in data−transfer phase is entered on the Protocol Spreadsheet via the SEND softkey labeled SND_DTA. This action is written to screen simply as SEND. CALL_SETUP_SEND cannot be executed in data−transfer phase.

   When Layer 1 is in data−transfer phase, a SEND action at Layer 2 will cause a transmission to go out onto the line automatically. No SEND action at Layer 1 is required.

EMULATE USER, USER CALLING AND CLEARING

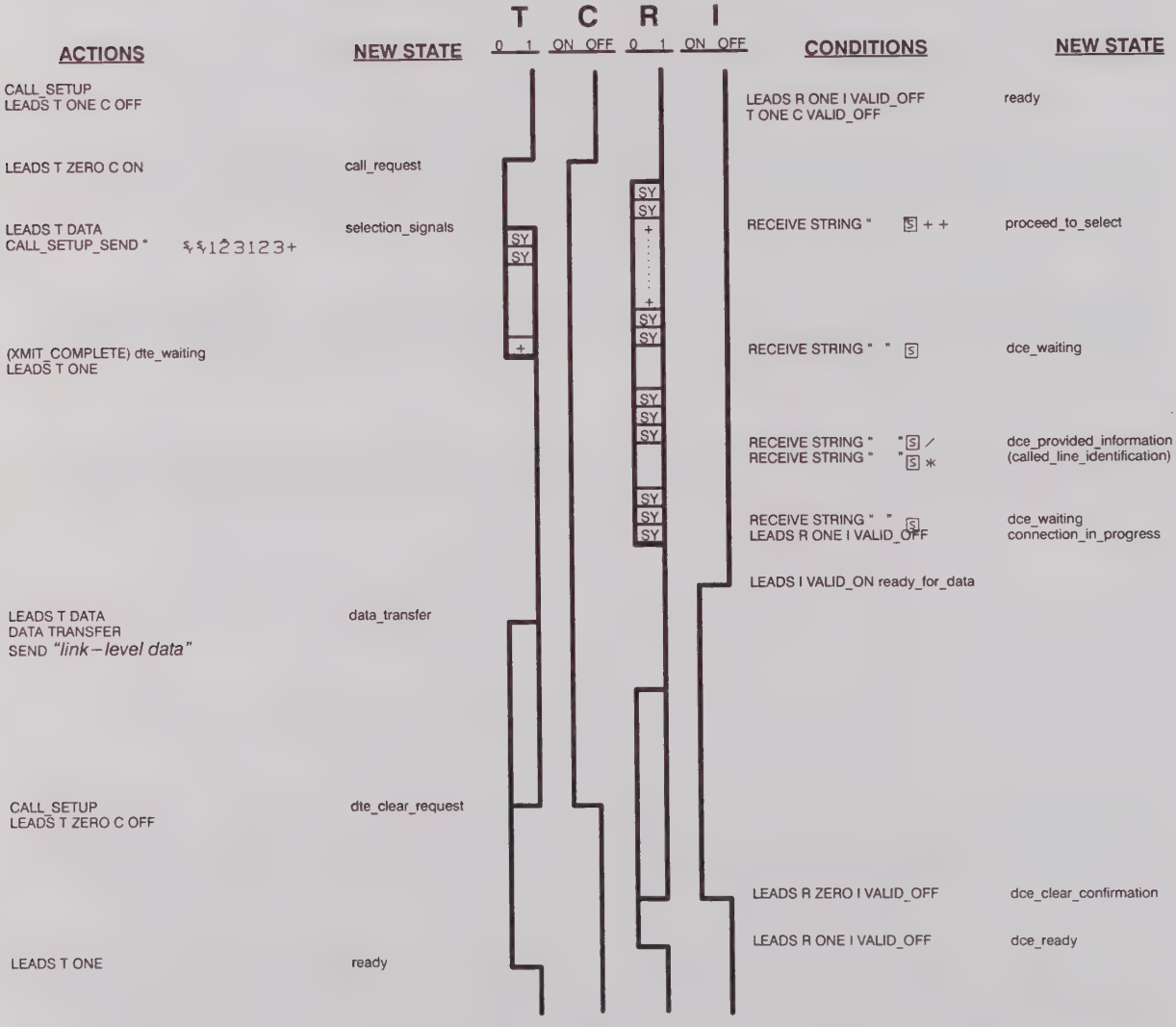


Figure 36-10 In this DTE–calling–and–clearing scenario, the INTERVIEW is on the user (DTE) side of the X.21 interface.

EMULATE SWITCH, USER CALLING AND CLEARING

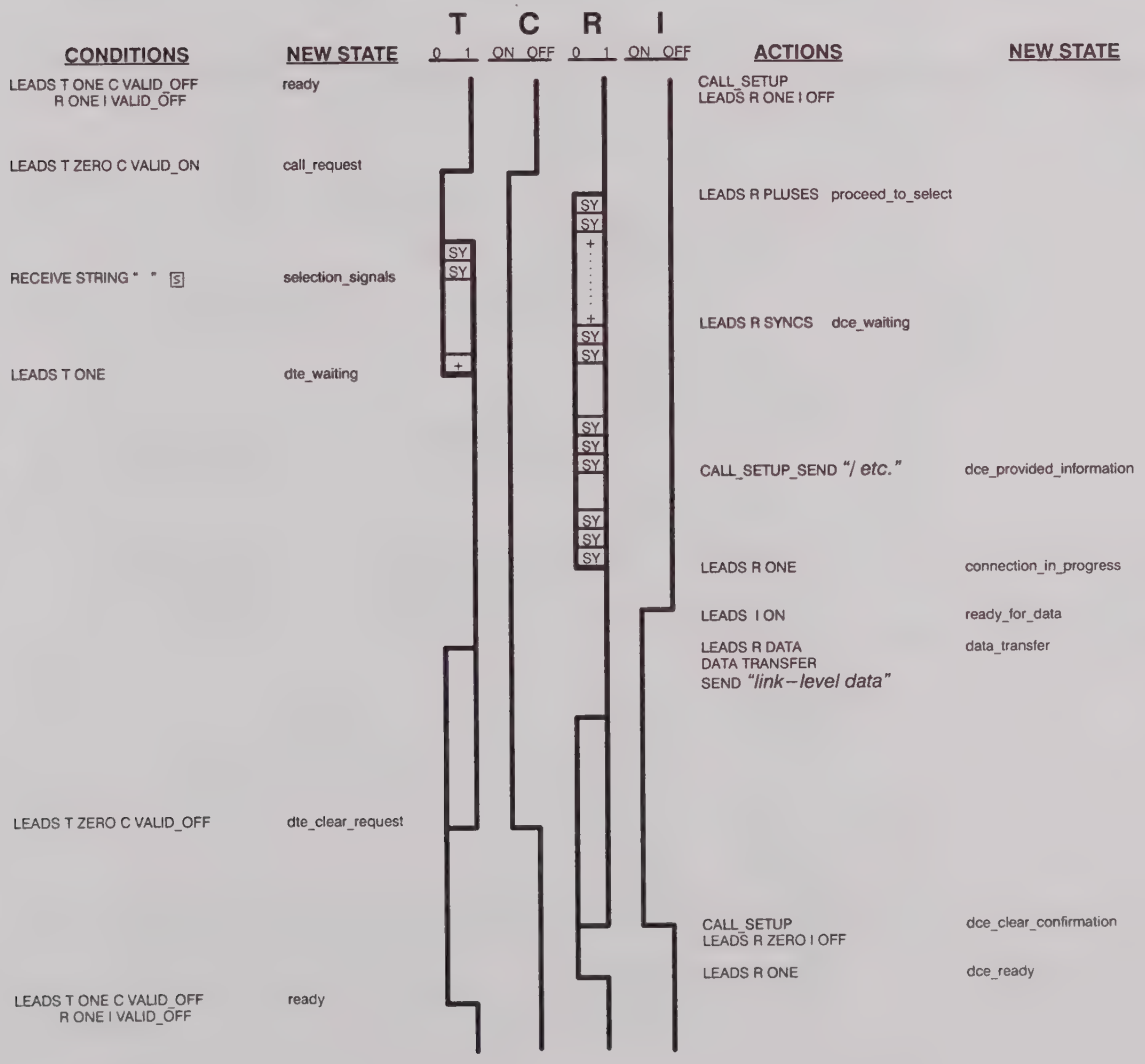


Figure 36-11 In this DTE–calling–and–clearing scenario, the INTERVIEW is on the network/switch side of the interface.

EMULATE USER, USER CALLED AND CLEARED

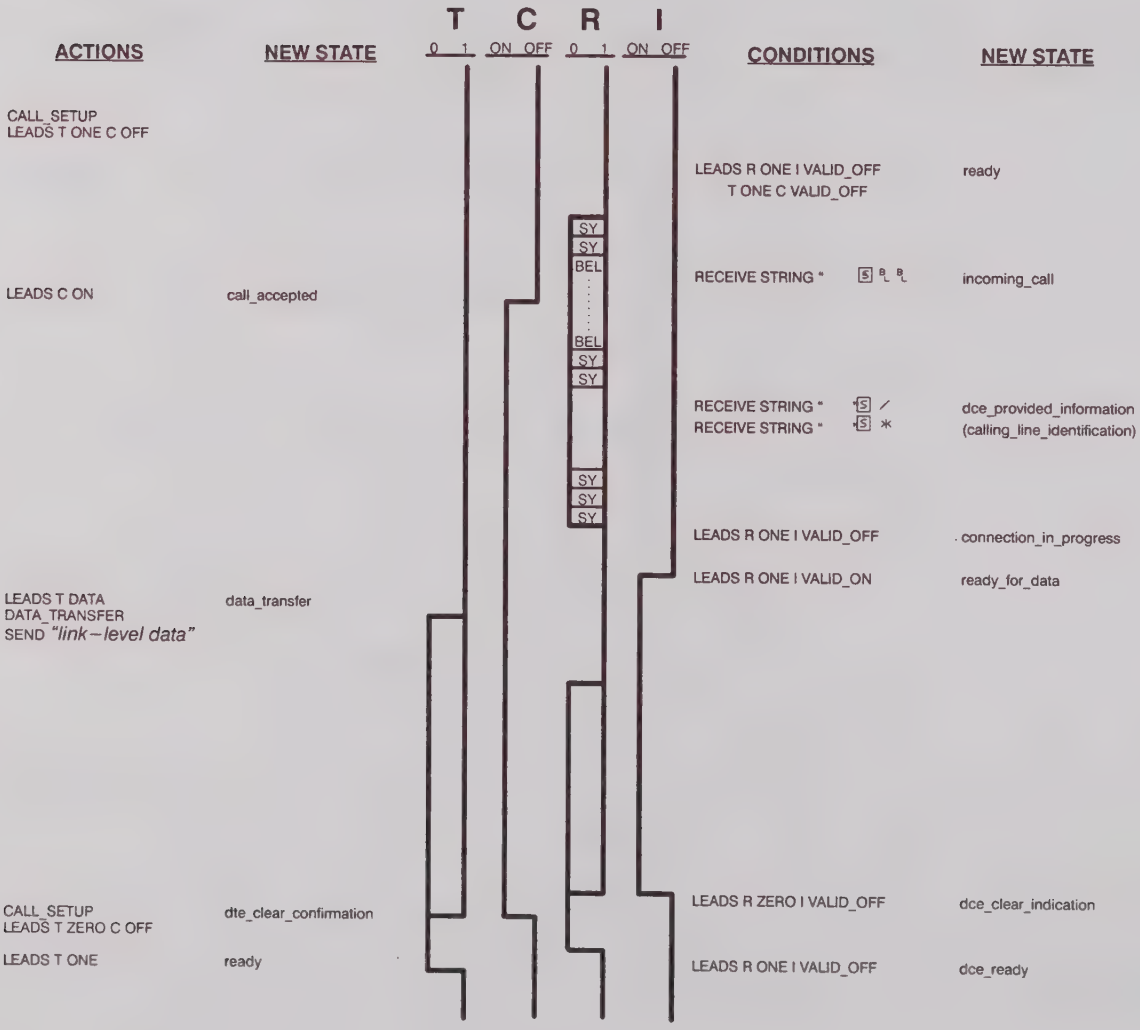


Figure 36-12 In this DTE–called–and–cleared scenario, the INTERVIEW is on the user (DTE) side of the X.21 interface.

EMULATE SWITCH, USER CALLED AND CLEARED

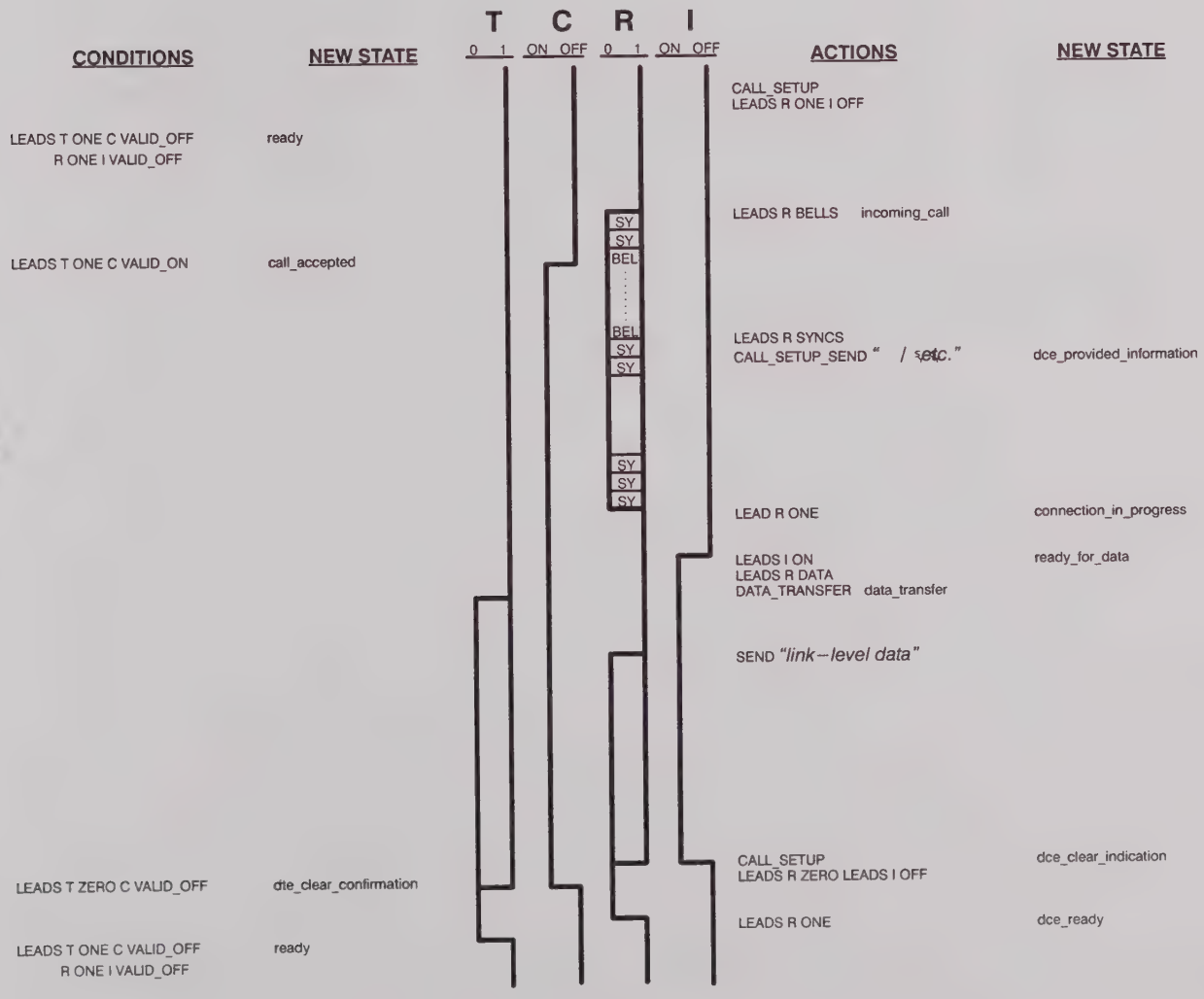


Figure 36-13 In this DTE–called–and–cleared scenario, the INTERVIEW is on the network/switch side of the interface.

37 X.25 Layer 2

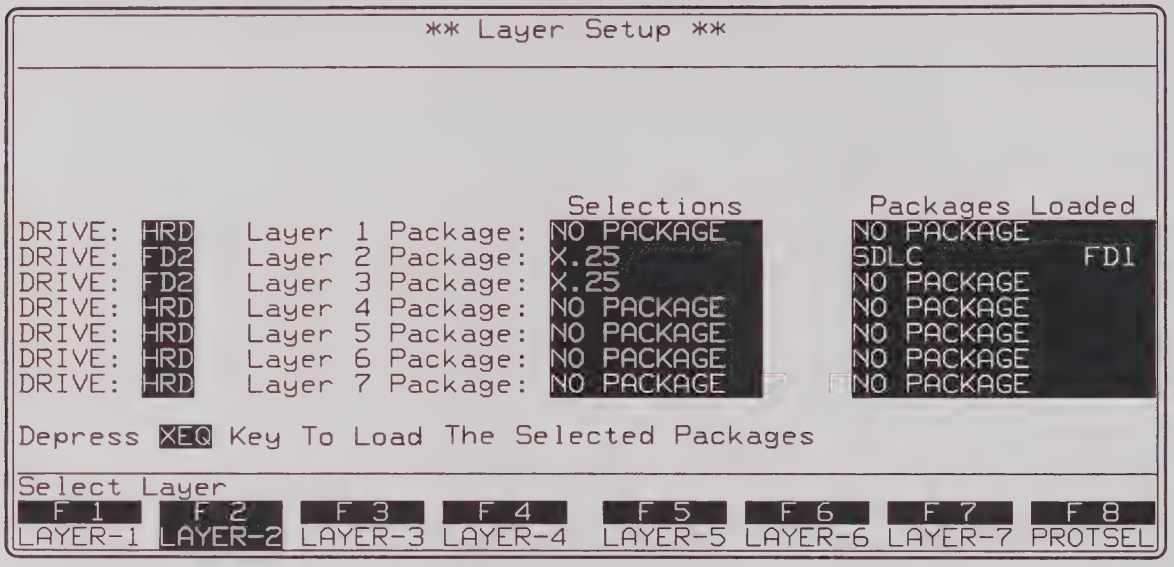


Figure 37-1 The X.25 personality package for Layer 2 is loaded from the Layer Setup screen.

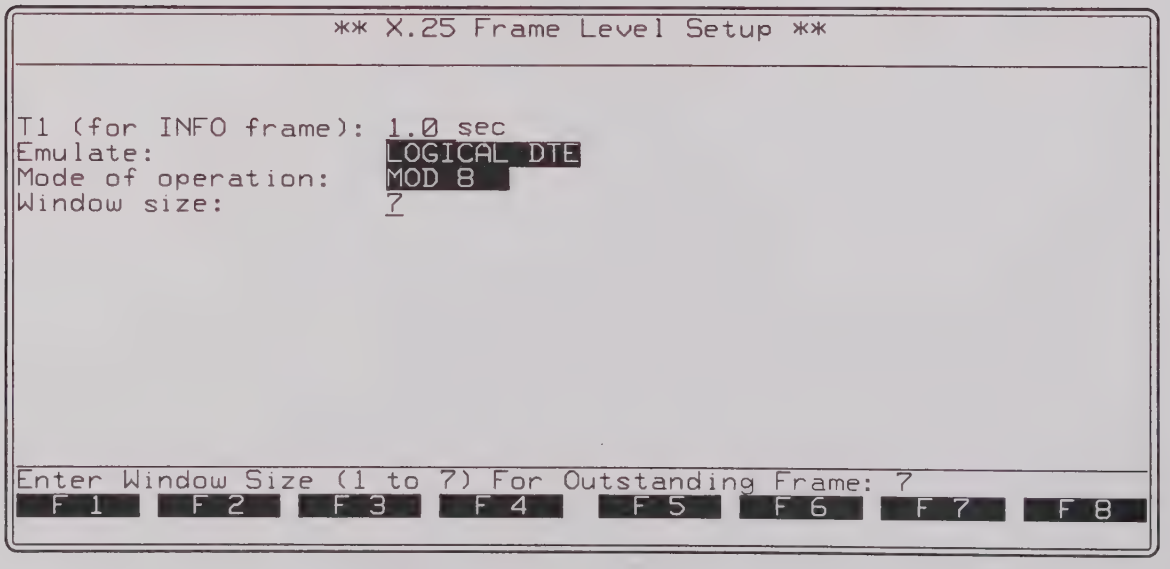


Figure 37-2 Protocol Configuration screen for X.25 Layer 2.

37 X.25 Layer 2

Layer 2 X.25 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 37-1 shows the Layer Setup screen configured to load in the Layer 2 X.25 package from floppy−disk Drive 2. Refer to Section 8 for details on operating the Layer Setup screen.

The Layer 2 X.25 package consists of the following:

- A special X.25 Frame Level Setup screen that controls certain parameters when the unit is tracing or emulating X.25.
- A protocol trace (illustrated in Figure 37-3) that distills from X.25 data the Level 2 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate X.25 programming. Figure 37-8 shows the softkey path to the first rack of condition softkeys when the X.25 package is loaded in at Layer 2.

37.1 Frame−Level Setup

The parameters on the X.25 Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the X.25 selection at Layer 2: **X.25** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up a prompt to **Select Protocol Configuration Screen**. Then press F2 (LAYER−2) to call up the X.25 Frame Level Setup screen.

The four parameter fields on this screen are shown in Figure 37-2. **T1**, **Emulate**, and **Window Size** apply to interactive (emulate) tests only. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) T1

Enter a four–digit (including decimal point) T1 timeout value in this field. The largest valid entry is 65.5 seconds. The smallest entry is .001 second, or 1 millisecond.

T1 is the name given to the retransmission timer for INFO frames. When a value is entered in the **T1** field on this menu, the Layer 2 package will handle T1 timings correctly, as follows:

- Whenever the INTERVIEW sends an I–frame at Layer 2 and *there are no previous frames sent by the INTERVIEW currently outstanding (unacknowledged)*, the timer starts timing down from the value entered on the Frame Level Setup screen.
- An acknowledgment by the device under test of the most recent frame transmitted by the INTERVIEW *stops* the timer (so that it does not expire).
- An acknowledgment by the device under test of a frame that is not the most recent frame transmitted by the INTERVIEW—an "incomplete" acknowledgment—restarts the T1 timer to the value selected on the configuration screen.

Expiration of this Frame Level Setup timeout can only be detected by a T1_EXPIRED condition on the Protocol Spreadsheet at Layer 2. This particular timeout cannot be detected by a generic condition of TIMEOUT T1.

According to the protocol, a T1_EXPIRED condition should result in a RESEND action.

(B) Emulate Logical DTE/DCE

There are two selections in the **Emulate** field on the X.25 Frame Level Setup screen, LOGICAL DTE and LOGICAL DCE. The entry in this field determines the Layer 2 address bytes used by the INTERVIEW during interactive testing.

Configured as a logical DTE, the INTERVIEW uses address $^{0}_{1}$ for commands and $^{0}_{3}$ for responses. Usually a logical DTE is the PAD at the user site.

Configured as a logical DCE, the INTERVIEW sends commands to address $^{0}_{3}$ and responds using address $^{0}_{1}$. Usually a logical DCE is a network switch.

Use the **Mode** selection (EMULATE DTE or EMULATE DCE) on the Line Setup menu to regulate the *physical* interface—whether to use Pin 2 or Pin 3 to transmit, and so on.

(C) Mode of Operation

The **Mode of Operation** field refers to the mode of numbering INFO and supervisory frames. There are two options, MOD 8 and MOD 128.

MOD 8 uses sequence numbers 0–7. MOD 128 adds an extra byte to the control field in INFO, RR, RNR , REJ, and SREJ frames. See Figure 37-5. This extra byte allows sequence numbers in a range of 0–127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate X.25 Layer 2 trace.

(D) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged I–frames that Layer 2 will buffer for retransmission. When the limit is reached, any further INFO frames that are named in SEND actions triggered at Layer 2 will be passed to Layer 1 for transmission but not buffered for retransmission.

The window is a queue that buffers frames for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) frame in the window. Successive RESENDs will send successive frames until there are no more frames to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

37.2 Protocol Trace

The Layer 2 X.25 package includes an automatic frame–trace display that summarizes link–level activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE to bring the protocol trace for X.25 Layer 2 to the screen. (If the X.25 package for Layer 3 is also loaded in, the L2TRACE softkey will appear after you have pressed PROTOCL, [F2] on the primary rack of display–mode softkeys.)

When running in High–Speed Frame Mode, more data could be passed to Layer 2 than there is room for in the buffer; this will cause an FEB overrun. If this happens, the error message **FE Buffer Overflowed – Some Frames Lost** will appear on the prompt line. The first time an FEB overrun occurs, an audible alarm will also sound; subsequent reccurrences will cause only the message to display (without any alarm). The trace will restart again but some data is lost with each occurrence.

Figure 37-3 is an example of the Layer 2 trace display. Each horizontal row in the trace represents a frame.

(A) The Protocol Trace in Freeze Mode

Press [FREEZE] to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing [ROLL FWD] or [↓] moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing [ROLL BACK] or [↑] moves the viewing window up to add a line of older data to the top of the screen.

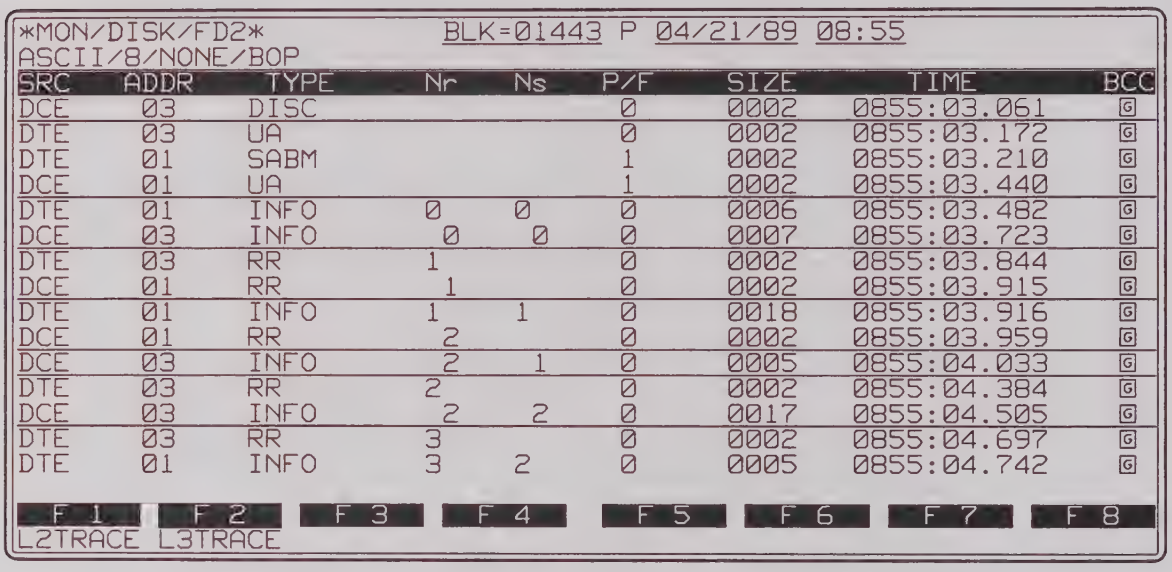


Figure 37-3 Each horizontal row on the trace display represents a frame.

Depression of the NEXT PAGE key adds fifteen lines—one full page—of newer frames to the frozen trace screen. Depression of PREV PAGE adds fifteen lines of older frames.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys. Figure 37-4 shows part of a dual–line data screen in Freeze mode. The first frame in the display is the same one that is traced at the top of Figure 37-3.

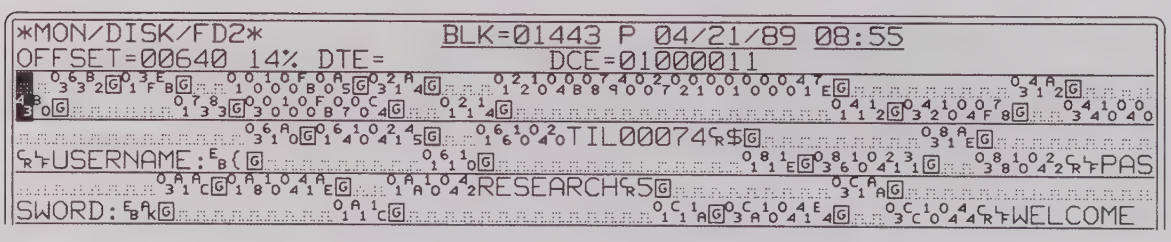


Figure 37-4 Data–display of Protocol Trace shown in Figure 37-3.

(B) Trace Columns

The columns in the protocol trace for Layer 2 X.25 are explained below.

1. *Source*. The **SRC** column identifies the lead on which the frame was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the frame, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

   Just as on the data display, RD data is underlined.

2. *Address*. The address octet (see Figure 37-5) is given in the **ADDR** column, with its two hexadecimal quartets presented as full−size alphanumerics. The address may be $^{0}_{1}$ or $^{0}_{3}$ in single−link operation.

   This column identifies the logical DTE and DCE. The logical DTE uses address $^{0}_{1}$ for *INFO frames and other command frames*, and address $^{0}_{3}$ for responses. The logical DCE uses address $^{0}_{3}$ for INFO frames and other commands, and address $^{0}_{1}$ for responses.

3. *Type*. The mnemonic (abbreviated) names for eleven frame types as they appear in the **TYPE** column of the protocol trace are shown in Figure 37-5 under "CONTROL." The control field, therefore, indicates the frame type. If a control octet does not fit any of the patterns in the figure, the frame is listed in the **TYPE** column as UNKWN followed by the hexadecimal value of the control byte: UNKWN=F3.

   If the number of bytes in the frame is below the required minimum, the frame is posted as INVALID.

4. *N(R) and N(S)*. One column on the frame−level trace is devoted to N(R) values, and one column to N(S). The frame types that include N(R) or N(S) fields in their control fields are indicated in Figure 37-5. N(R) and N(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

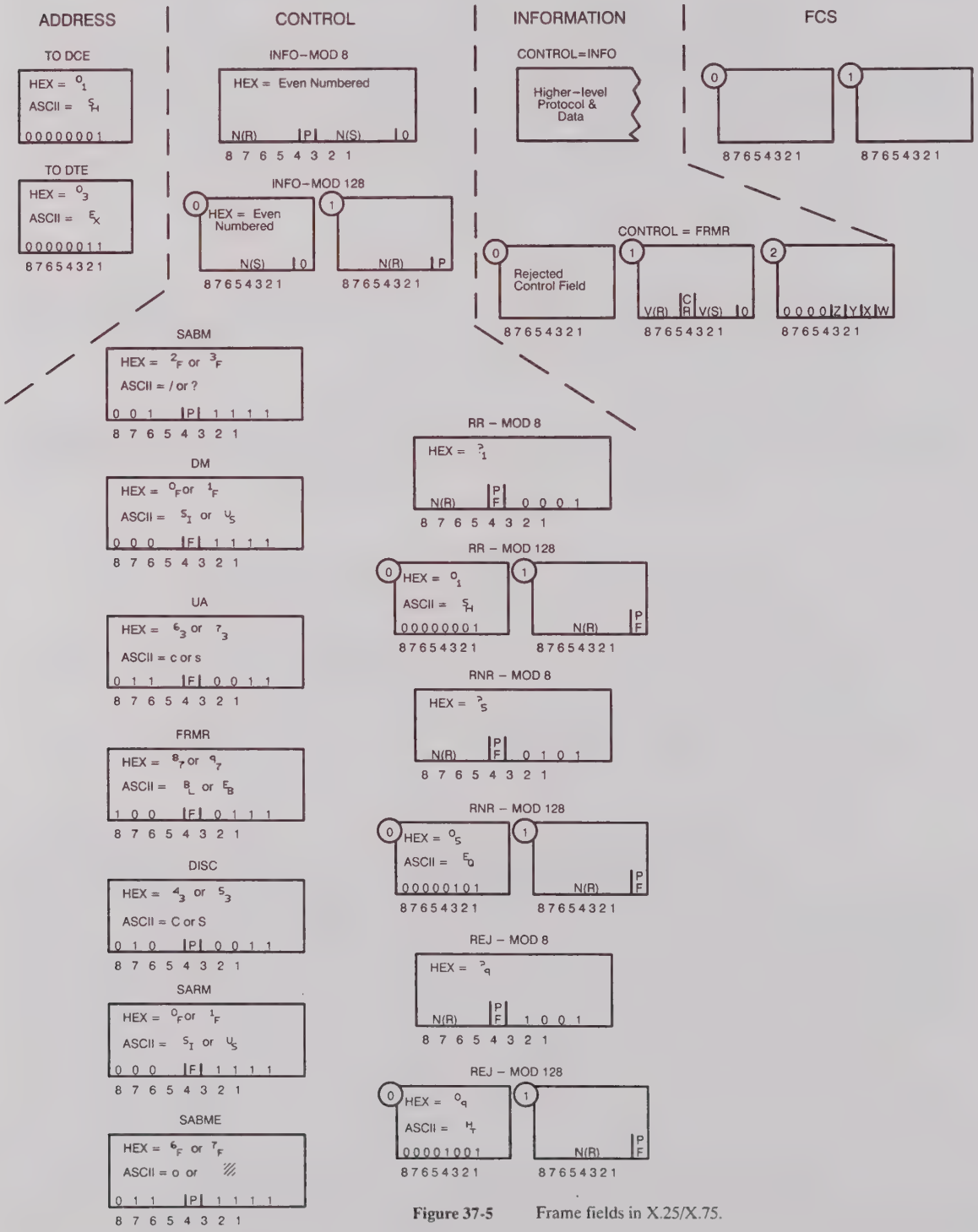


Figure 37-5 Frame fields in X.25/X.75.

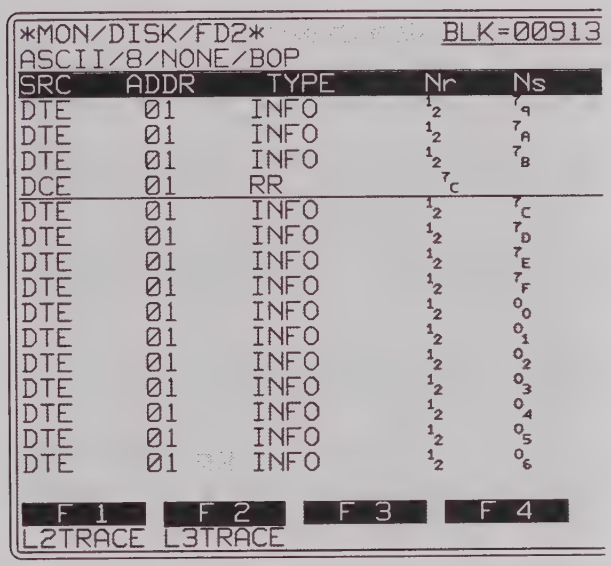


Figure 37-6 MOD 128 sequence numbers are displayed in two–digit hexadecimal *characters*.

N(R) and N(S) values are presented in decimal format in modulo–8 traces. Each column displays a single digit that represents a 3–bit binary value. For modulo 128, the values % to 7F are given in "character" format, where the columns contain a two–digit hexadecimal character (see Figure 37-6).

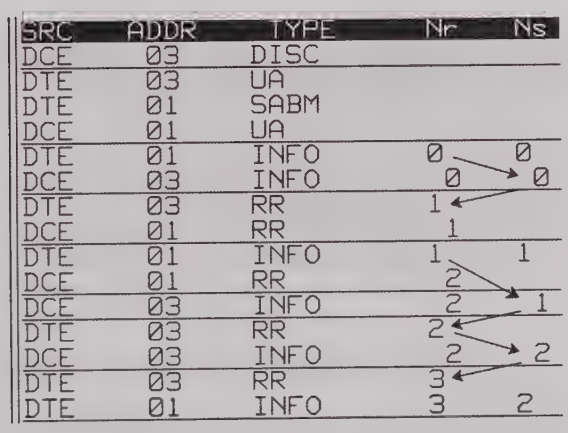


Figure 37-7 Nr and Ns columns are staggered, with the *outside* columns representing the sequence of *DCE* numbered I–frames.

Note that the **Nr** and **Ns** columns on the trace are staggered to suggest four columns. The two outside columns, comprised of the DTE's N(R) value and the

DCE's N(S), form a numbering sequence for DCE I–frames. The arrows in Figure 37-7 indicate the sequence: the DTE expects to receive frame 0, the DCE sends frame 0. The DTE expects frame 1; it asks for frame 1 again; finally the DCE sends frame 1. And so on.

The two inside columns reveal a similar pattern for DTE I–frames.

5. *P and F.* The status of the poll or the final bit is given in the **P/F** column. Whether this bit is the P or F bit is indicated for most frame types in Figure 37-5 (under "CONTROL").

   The setting of the P bit in an INFO frame often denotes the retransmission of an unacknowledged frame following a T1 timeout.

6. *Size.* The number of bytes in each frame is given in this column in four decimal digits. The count begins with the address byte and excludes the two–byte FCS. Frames without I–fields show a count of two.

7. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

   If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

8. *Frame checking.* An X.25 frame ends as soon as a $^{7}_{E}$ flag or seven 1−bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

   A for abort is posted to the display when a frame is ended by seven 1−bits.

37.3 Monitor Conditions

When the Layer 2 X.25 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 37-8.

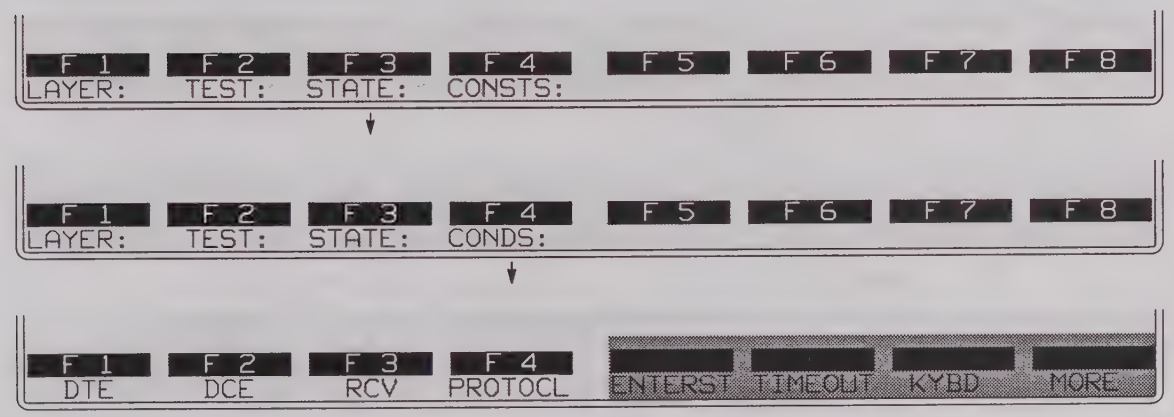


Figure 37-8 Unlike *RCV* conditions, the softkeys for *DTE* and *DCE* are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of frames: INFO, SNRM, UA, and so forth.

(A) Frame Types

The softkeys for INFO, supervisory, unnumbered, and "other" frames are illustrated in Figure 37-9. Press a softkey to write one of these frame types to the Layer 2 spreadsheet. DTE or DCE followed by a frame−type mnemonic—DTE INFO, for example, or DCE SNRM—is a complete condition and will come true if a matching frame is monitored. Address, poll/final, and BCC conditions may be added to the simple frame mnemonic, but they are optional.

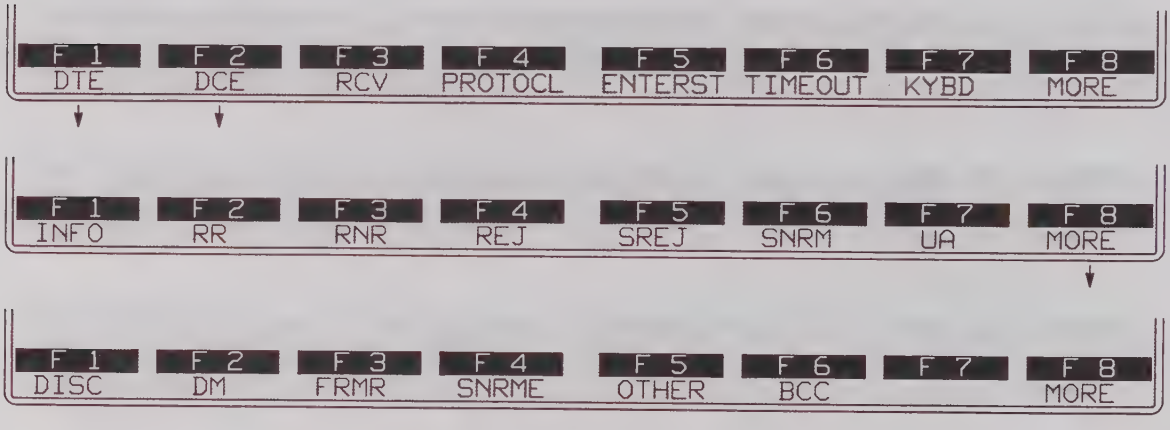


Figure 37-9 Frame types.

1. *Info frames*. INFO frames differ for MOD 8 and MOD 128 numbering schemes. (See Figure 37-5.) For spreadsheet conditions to match I−frames accurately, the correct numbering system ("**Mode of Operation**") should be selected on the Frame Level Setup screen.

2. *Supervisory frames*. The four supervisory−frame types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), REJect, and SREJ (Selective Reject). These frames always contain N(R) fields (see Figure 37-5) and serve mainly to acknowledge or reject INFO frames.

   Like INFO frames, supervisory frames are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

3. *Unnumbered frames*. Unnumbered frames generally assist in link−setup and takedown. Different sset−mode commands are used in different protocols: SABM for LAPB MOD 8 and SABME for LAPB MOD 128.

4. *Other frames*. Any frame type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. See Figure 37-10. Then enter the hex byte in the form of two alphanumerics. Here, for example, is a SABM (with the P−bit set) entered as a hexadecimal:

   CONDITIONS: DCE OTHER 3F

   Address, poll/final, and BCC conditions may be appended to OTHER conditions. In MOD 8, the P/F bit is already specified in the hex entry, and a P/F condition will be ignored.

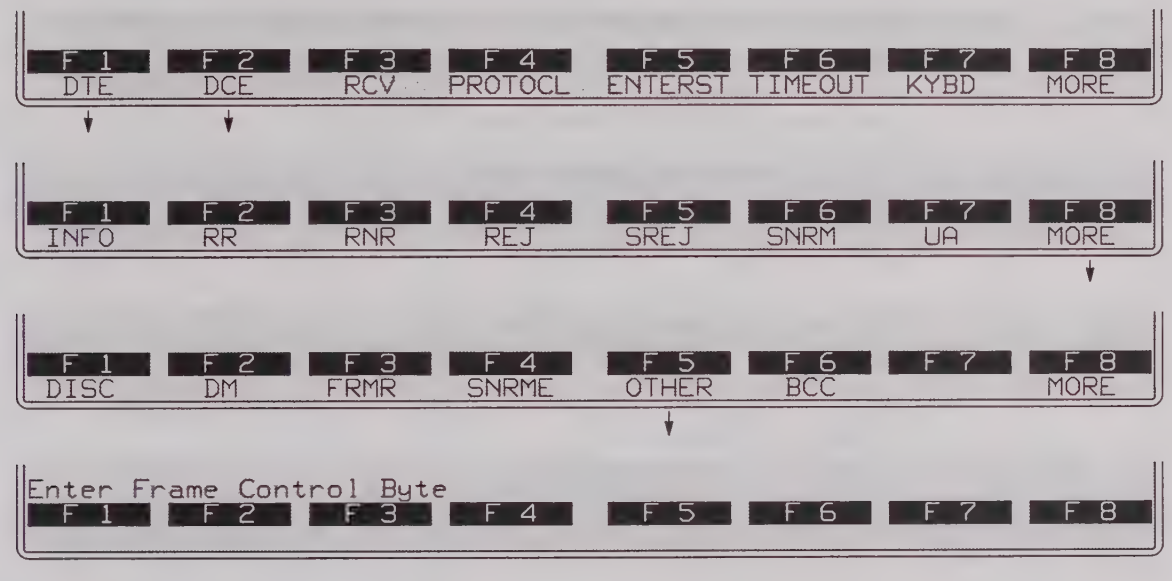


Figure 37-10 The hex value of any frame may be specified under *OTHER*.

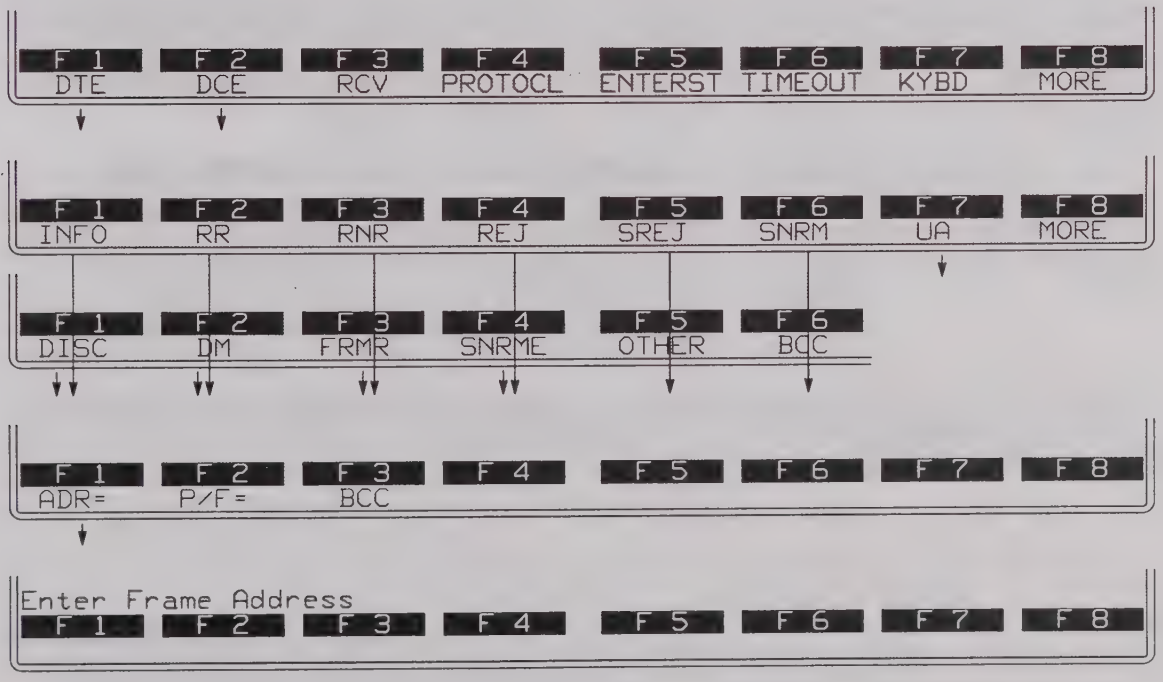


Figure 37-11 The hex value of the address byte is entered as two alphanumerics for *all* frame types.

5. *Address*. An address condition may be added to INFO, supervisory, unnumbered, and OTHER conditions. Press the softkey for ADR=, shown in Figure 37-11. Then enter the hexadecimal address octet as two alphanumerics. The address octet $^{0}_{1}$, for example, appears as follows:

   CONDITIONS: DTE INFO ADR= 01

   To bypass the ADR= selection (as well as the other options on the same rack of softkeys in Figure 37-11) press [DONE].

6. *Poll/final bit*. P/F conditions are optional for all frame types. P/F values of 0 or 1 are entered by the softkey sequence in Figure 37-12.

   Press [DONE] to bypass the P/F= condition and the other conditions on the same softkey level in Figure 37-12.

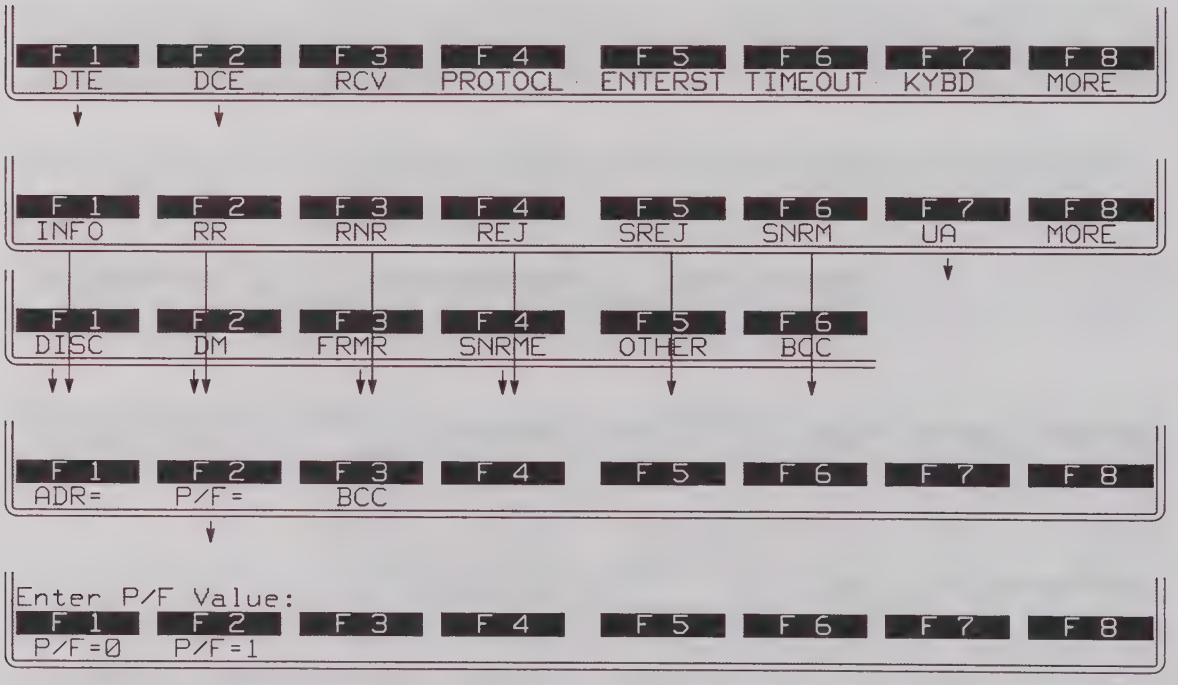


Figure 37-12 The value of the P/F bit may be chosen as a condition.

(B) BCC Conditions

DTE and DCE frames may be monitored for good and bad frame checks and for aborts. *All* DTE or DCE frames may be monitored with respect to frame checking, as in this example:

CONDITIONS: DTE BDBCC

The softkey sequence for this spreadsheet entry is given in Figure 37-13.

Or a particular *type* of frame may have a BCC or abort condition appended to it:

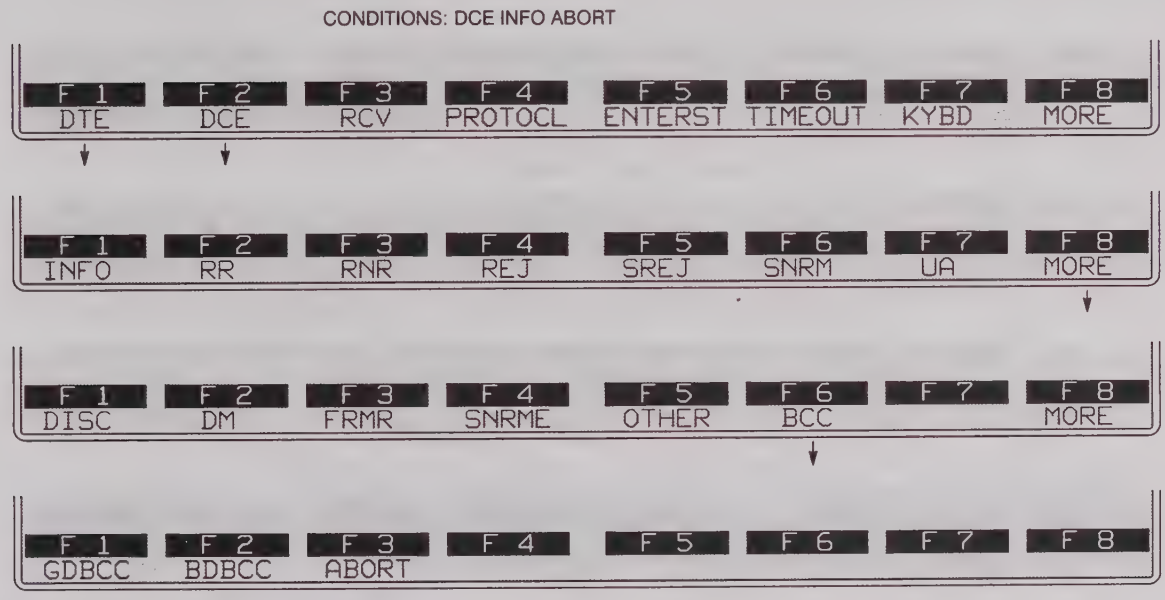


Figure 37-13 A condition may search for *all* good, bad, or aborted frames.

37.4 Emulate–Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for **Mode:** EMULATE DTE or EMULATE DCE.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data lead for X.25 frame types. RCV conditions operate only in emulate modes, and they check only the data lead that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV INFO P/F=1 looks the same as DCE INFO P/F=1—there are important differences that are noted below.

1. *Valid frame sequencing.* To satisfy RCV conditions, numbered frames must have correct N(R) and N(S) sequencing.

2. *Good BCC.* RCV conditions cannot match frames with bad frame checks, nor can they match aborted frames. (Emulate–mode conditions are designed for ease of programming, and the assumption is that as an X.25 emulator, you are not required to acknowledge—or negative–acknowledge—bad or aborted frames.)

If you wish to count bad BCCs or aborts, use DTE or DCE conditions instead of RCV conditions.

3. *Address in supervisory frames.* An address condition *must* be added to a n RCV RR, RCV RNR, RCV REJ, or RCV SREJ condition. (In a DTE or DCE supervisory condition, the address is optional.) The address may be entered as in DTE/DCE conditions:

   CONDITIONS: RCV RR ADR= 01

   Or the address may be entered simply as COMMAND or RESP—RCV RR RESP, for example. COMMAND and RESP conditions will look for a specific address, 03 or 01, depending on the selection in the **Emulate** field on the X.25 Layer 2 Setup screen (see Section , above). A logical DTE will *receive* commands addressed to 03, and it will receive responses that have the address 01. A logical DCE receives commands addressed to 01 and responses that use 03.

   CMND and RESP softkeys are shown in Figure 37-14.

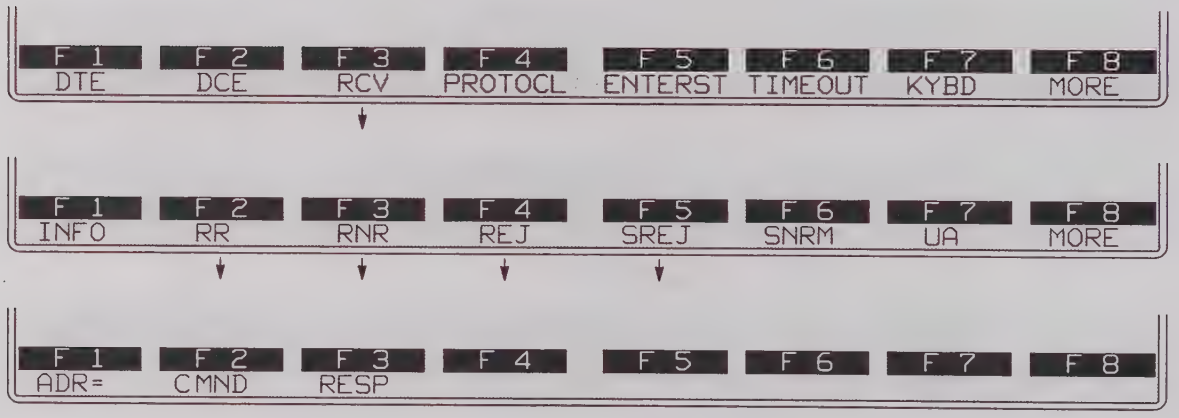


Figure 37-14 Addresses are required in *RCV RR*, *RCV RNR*, *RCV REJ*, and *RCV SREJ* conditions.

4. *Type invalid.* RCV conditions can detect frames that are invalid "types"—the control field is missing, for example, or the I–field is missing in an I–frame. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 37-15.

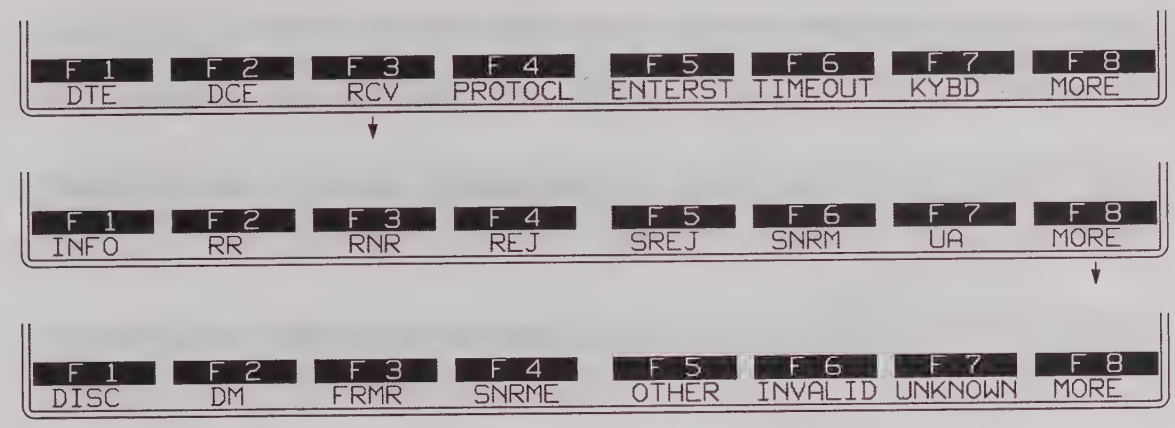


Figure 37-15 *INVALID* and *UNKNOWN* are frame types for *RCV* conditions.

5. *Type unknown*. A frame may be valid in all respects but have a control field that indicates a nonstandard frame type. Such a frame may be matched by a RCV UNKNOWN condition (Figure 37-15).

(B) N(S) Error

As a Layer 2 emulator, you do respond to INFO frames that have N(S) errors. These are detected as NS_ERR conditions, not as RCV INFO conditions.

NS_ERRs apply only to frames received when you are emulating. The same frame that triggers an NS_ERR condition also may satisfy a DTE INFO or DCE INFO condition—but not a RCV INFO condition.

NS_ERR will come true for any received frame whose N(S) value is not one higher than the previous N(S).

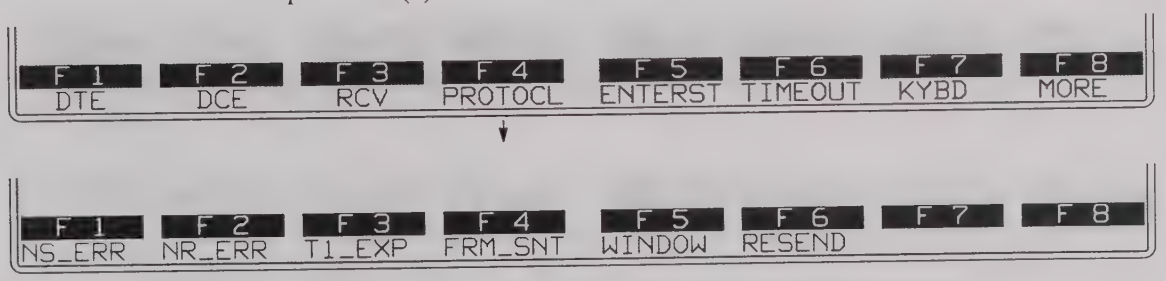


Figure 37-16 The *PROTOCL* key brings up six SDLC emulate conditions.

In the first rack of condition softkeys at Layer 2, press PROTOCL. Then press the softkey for NS_ERR. See Figure 37-16.

(C) N(R) Error

Received INFO or supervisory frames may have N(R) errors. Such errors are detected as NR_ERR conditions, not as RCV INFO or RR (or RNR or REJ) conditions.

A valid N(R) is any value that (1) acknowledges a frame that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other N(R) value is detected as an error.

(D) T1 Expired

This condition detects the expiration of the T1 timeout–timer that is regulated on the X.25 Frame Level Setup screen. See Section 37.1(A), above.

(E) Frame Sent

This condition is true when, as a result of a SEND or RESEND action, a frame has been passed down to Layer 1. Note that merely SENDing a frame *does not actually transmit the frame onto the line* if, for example, Layer 1 is the X.21 package in call–setup phase.

(F) Window Conditions

The size of the Layer 2 retransmit window is configured on the X.25 Frame Level Setup screen. See Section 37.1(D). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 37-17.

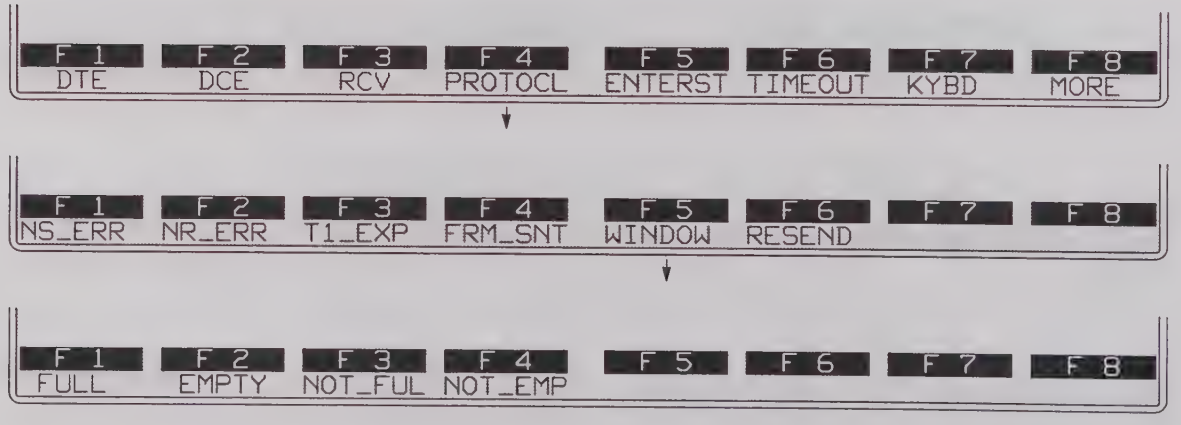


Figure 37-17 When the retransmit window fills, Layer 2 stops buffering frames for retransmission.

WINDOW FULL is true when the window is full of unacknowledged frames and the Layer 2 protocol package will not buffer additional frames until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no frames outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR RESP
   WINDOW NOT_EMPTY
```

CAUTION: Window conditions are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as a RCV condition.*

(G) More to Resend

Frames in the window may have to be resent, usually as the result of an idle–timer timeout or a Reject frame. One RESEND action retransmits one frame in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent frames. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
   CONDITIONS: RCV REJ RESP
   NEXT_ST: recover
STATE: recover
   CONDITIONS: ENTER_STATE
   ACTIONS: RESEND FIRST
   CONDITIONS: FRAME_SENT
      MORE_TO_RESEND
   ACTIONS: RESEND NEXT
   CONDITIONS: FRAME_SENT
      NO_MORE_TO_RESEND
   NEXT_ST: xfer
```

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 37-18.

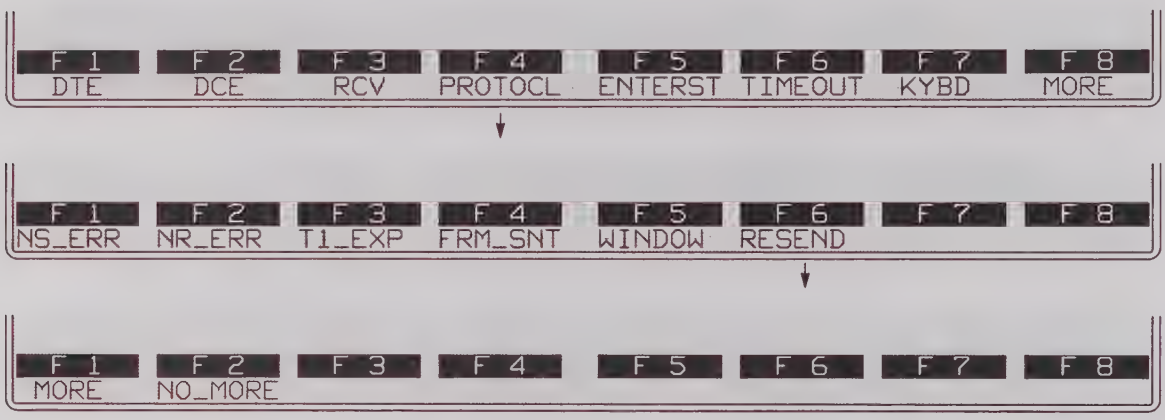


Figure 37-18 The *MORE_TO_RESEND* condition allows you to resend the entire window of frames and then stop when there are *NO_MORE_TO_RESEND*.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as FRAME_SENT.*

37.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 2, press DONE to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the X.25 Layer 2 personality package are shown in the racks of softkeys in Figure 37-19. Except for ENHANCE and SUPPRES, the actions shown have meaning only when the INTERVIEW is emulating DTE or DCE, and not when it is monitoring the line passively.

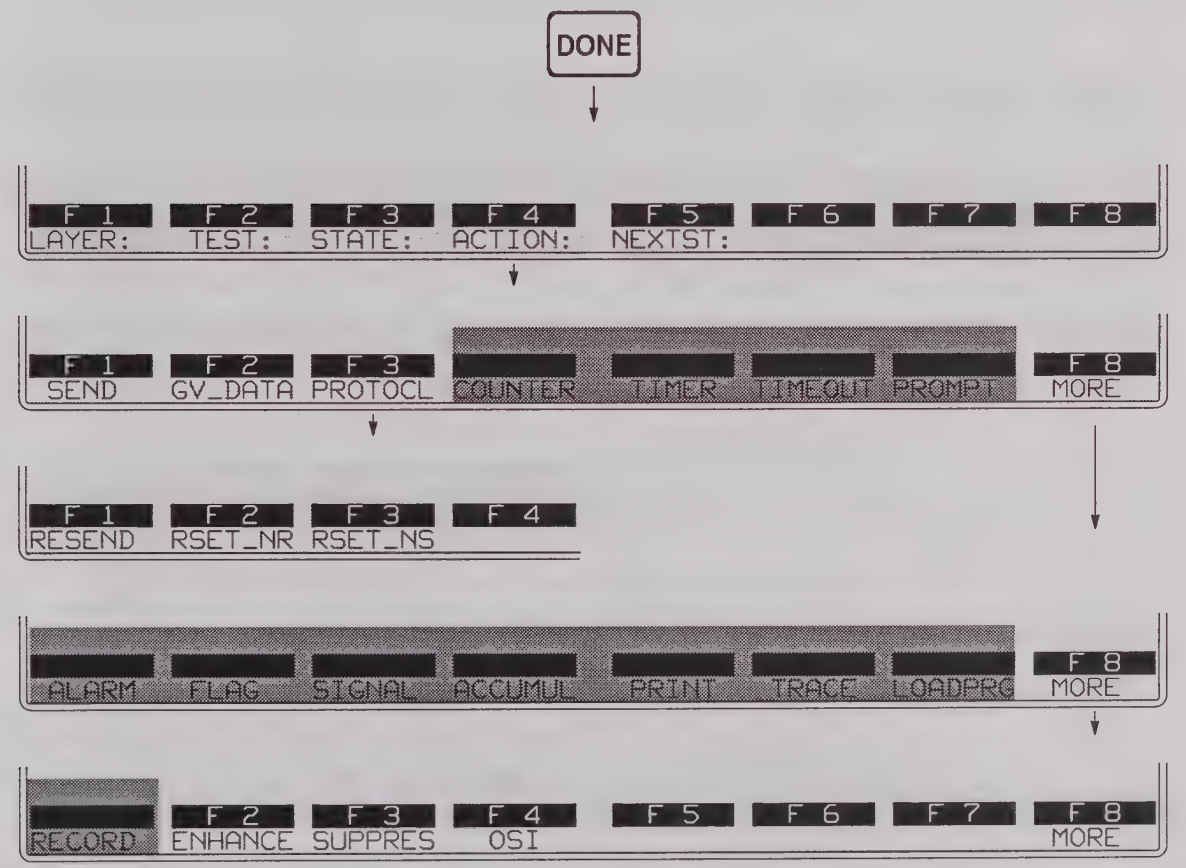


Figure 37-19 Action softkeys specific to X.25 Layer 2.

(A) Send Actions

Press the softkey for SEND to access two racks of softkeys with names of frame types that may be named in SEND actions. All data generated by the Layer 2 X.25 package must be enclosed in a frame that is identified in a SEND action by *type*. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of frame types is given in Figure 37-20.

When conditions are true for a SEND action, frames are sent immediately down to Layer 1 to be transmitted there. (Note that when the X.21 Layer 1 package is loaded, the sending down of INFO frames will be conditional on data–transfer phase being active at Layer 1.)

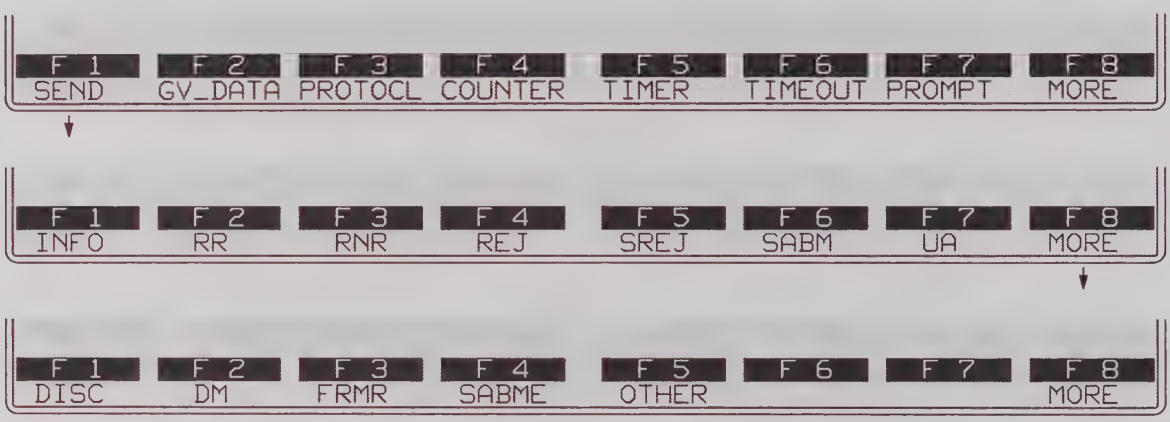


Figure 37-20 *SEND* actions always specify a frame type.

1. *INFO frames*. SEND INFO is a complete action–entry. Address, poll bit, N(R), N(S), string, and BCC parameters may be added to an INFO frame, but they are optional. If no optional parameters are entered, the INFO frame will default to the following parameters:

   - The address will be appropriate for an INFO frame sent by the "logical emulator" selected on the Frame Level Setup screen. See Section 37.1(B).
   - The poll bit will be set to 1.
   - The N(R) will increment to the "automatic" value, one higher than the last valid N(S) received.
   - The N(S) will increment to the "automatic" value, one higher than the last valid N(S) sent.
   - Since there is no default data–string, the I–field will be empty.
   - The BCC will be good.

   The default parameters for INFO and other frames are given in Table 37-1.

   If a Layer 3 package is installed and Layer 3 data is being handed down to Layer 2, the following condition–and–action trigger will accept this data and convey it properly to Layer 1:

```
CONDITIONS: DL_DATA REQ
ACTIONS: SEND INFO " ((DL_DATA)) "
```

SEND INFO actions pass the INFO frame immediately to the next layer down. If the retransmit window is full, the frame is still sent—but it is not buffered in the window and cannot be *re*sent.

An INFO frame will be buffered for retransmission regardless of the status of the window if a specific value is entered for the NS= parameter (see "*N(S)*," below). The specific N(S) value will clear the window and the INFO frame will be buffered in the first window position.

Table 37-1
Default Parameters in *SEND* Actions

SEND	ADR= logical DTE	logical DCE	P/F	NR=	NS=	STRING	BCC
INFO	01	03	1	AUTO	AUTO	none	GDBCC
SABM	01	03	1	N/A	N/A	N/A	GDBCC
UA	03	01	L	N/A	N/A	N/A	GDBCC
DISC	01	03	1	N/A	N/A	N/A	GDBCC
DM	03	01	L	N/A	N/A	N/A	GDBCC
FRMR	03	01	L	N/A	N/A	none	GDBCC
SABME	01	03	1	N/A	N/A	N/A	GDBCC
RR CMND	01	03	1	AUTO	N/A	N/A	GDBCC
RR RESP	03	01	L	AUTO	N/A	N/A	GDBCC
RNR CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
RNR RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
REJ CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
REJ RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
SREJ CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
SREJ RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
OTHER	01	03	N/A	N/A	N/A	none	GDBCC

2. *Unnumbered frames*. SEND SABM, SEND UA, and so forth, are complete action–entries. Address, P/F–bit, string, and BCC parameters may be added to the SEND action, but they are optional. Default values are sent in the absence of specific optional entries: see Table 37-1.

3. *Supervisory frames*. An address value must be added to SEND RR, SEND RNR, SEND REJ, and SEND SREJ actions. The address may be entered as a specific value.

```
ACTIONS: SEND RR ADR= 03
```

Or the address may be entered simply as CMND or RESP—SEND RR RESP, for example. CMND and RESP frames will carry address 01 or 03, depending on the selection in the **Emulate** field on the X.25 Frame Level Setup screen . See Section 37.1(B). Refer to Table 37-1 for the address values sent by the two different logical emulators.

Figure 37-21 shows the address selections for all supervisory frames.

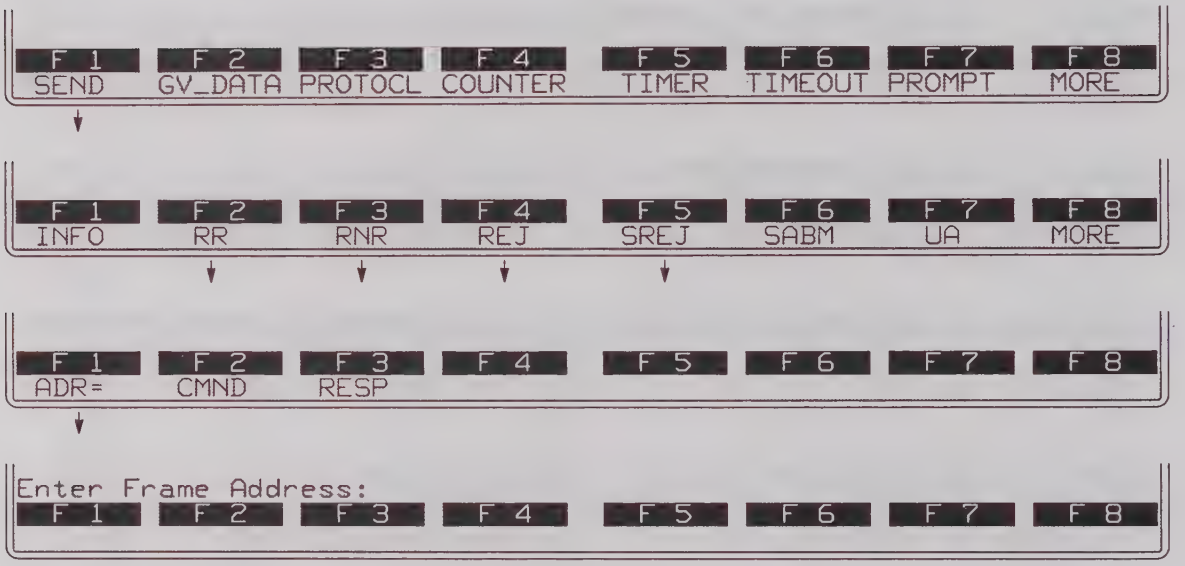


Figure 37-21 An address value must be added to *SEND RR*, *SEND RNR*, *SEND REJ*, and *SEND SREJ* actions.

4. *Other frames*. Any frame type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 37-20. Enter the hex value in the form of two alphanumerics. Then press the softkey for ADR= (Figure 37-22) and enter an address value. Here is a DISConnect command entered as a SEND OTHER action:

   ACTIONS: SEND OTHER 43 ADR= 03

   P/F, N(R), and N(S) fields are implied in the user–entered hexadecimal control field. An address *should* be added to a SEND OTHER action, but if it is not, the default is the (CMND) address 01 when the **Emulate** field on the X.25 Frame Level Setup screen shows LOGICAL DTE . The address defaults to 03 for LOGICAL DCE . The other default parameter is a good BCC. (In MOD 128, P/F is not included in the hex entry and is a valid optional entry.)

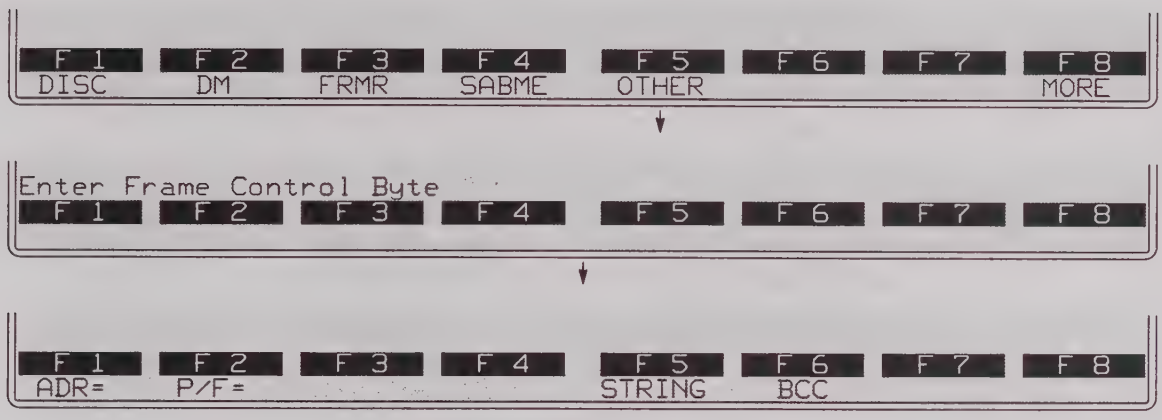


Figure 37-22 *SEND OTHER* actions always specify a *type* value.

5. *Address*. An address may be specified for INFO, unnumbered, and OTHER frames. It must be specified for supervisory frames. There are three softkeys pertaining to address in supervisory frames, ADR=, CMND, and RESP. See under "*Supervisory frames*," above.

   The ADR= entry is always followed by the hexadecimal address octet typed as two alphanumeric characters. The address field $^{0}3$, for example, appears as follows:

   ACTIONS: SEND RR ADR= 03

6. *Poll/final bit*. The P/F−bit is an optional entry in all SEND actions. P/F values of 0, 1, or LOOPBAK are entered by the softkeys in Figure 37-23. If P/F= LOOPBACK, the bit will echo the last P/F bit received. (Looping the P/F−bit is appropriate for UAs and supervisory frames.) Default P/F values are given in Table 37-1.

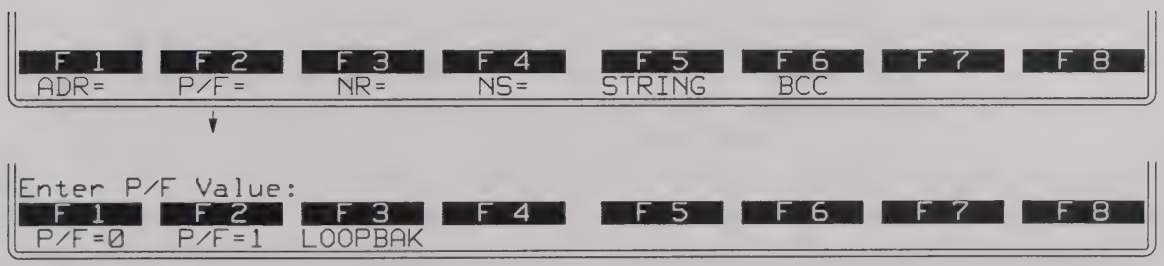


Figure 37-23 A P/F value is optional in all *SEND* entries.

7. *N(R)*. N(R) fields are transmitted in INFO and supervisory frames.

   To specify an N(R) value, press the softkey for NR= (see Figure 37-24). Enter a hexadecimal value written as one or two alphanumeric digits. For example, an entry that represented the highest valid N(R) in MOD 8 would be NR= 7. The highest valid entry in MOD 128 would be NR= 7F.

   Other N(R) options are ACK_NS, LAST_NR, and AUTO. (See Figure 37-24.) ACK_NS means that your N(R) will acknowledge (that is, it will be *one higher than*) the last N(S) value you received. Normally this will be the correct N(R), except in cases where the last N(S) received was erroneous. The NR= ACK_NS selection allows you to overlook N(S) errors.

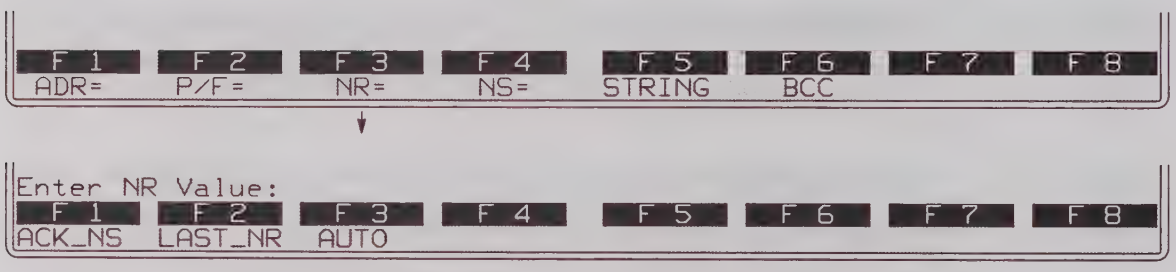


Figure 37-24 The N(R) field may be specified in INFO and supervisory frames to be sent.

LAST_NR means that you simply repeat the last N(R) you sent. Normally this is the correct N(R) following a bad N(S). The NR= LAST_NR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal X.25 Layer 2 station, ACKing valid N(S) values and repeating your last N(R) whenever an invalid N(S) is received. AUTO is the default N(R) selection in SEND INFO, SEND RR, SEND REJ, and SEND SREJ actions. See Table 37-1.

8. *N(S)*. N(S) fields are transmitted in INFO frames only. (See the frame–field diagrams in Figure 37-5.) Entries for N(S) in SEND INFO actions are optional. The softkeys that open below NS= are illustrated in Figure 37-25.

   To specify an N(S) value, press the softkey for NS=, then enter a hexadecimal in the form of one or two alphanumerics. Valid hex entries are the same as for N(R). A SEND INFO action that *specifies* an N(S) value—NS= 0, for example—will clear the window so that the INFO frame is buffered immediately.

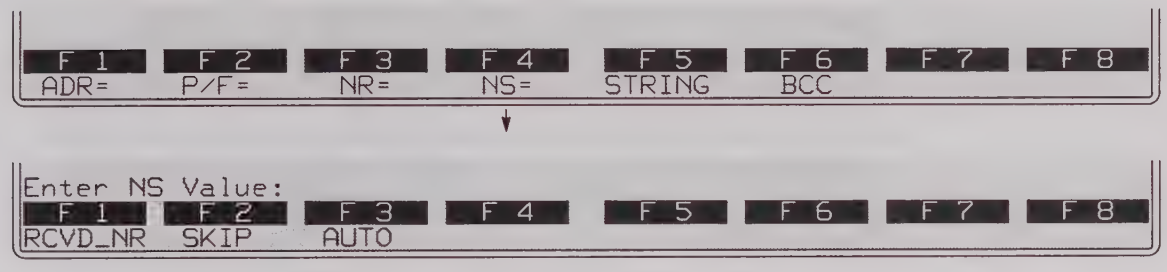


Figure 37-25 The N(S) field may be specified in a *SEND INFO* action.

Other N(S) options are RCVD_NR, SKIP, and AUTO. RCVD_NR means that you send the N(S) value that the other side says it is expecting. This is the valid N(S) in most cases, but not when you send two or more I−frames in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct N(S). This will look to the other side as though the line has taken a "hit" and a frame has been lost. This selection causes the window to be cleared.

NS= AUTO is the default setting for SEND INFO actions. AUTO means that every *new* INFO frame sent will have an N(S) value of one higher than the previous.

9. *String*. Strings are sent at X.25 Layer 2 only as adjuncts to frame−types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

   Press the SEND softkey followed by the softkey for a frame type. Add any necessary or desired SEND options for the particular frame type. Then press the STRING softkey (Figure 37-25).

   There is no spreadsheet keyword that identifies send−strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " −character in your string, type \" .) See Section 33 for more information on strings.

   Here is a simple SEND action that includes no options besides a string:

   ACTIONS: SEND FRMR " $^{1}_{1}{}^{6}_{6}{}^{0}_{8}$ "

   And here is a SEND action that includes a full complement of optional fields, including a string:

   ACTIONS: SEND INFO ADR= 03 P/F= 0 NR= AUTO NS= AUTO
   " $^{1}_{0}{}^{0}_{2}{}^{0}_{0}$ This is user data." GDBCC

   Most ASCII−keyboard, control, and hexadecimal characters are legal in a send−string. Special keys (FLAG, DON'T CARE, BIT MASK) are not legal. Refer to Table 33-2.

   To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)) . Remember that the double parens are special

characters produced by the CTRL–9 and CTRL–0 combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

10. *BCC.* There are three BCC options for every SEND action at X.25 Layer 2. One of the options, GDBCC, is the default. Any frame that does not request a bad BCC or an abort will have a good frame–check sequence calculated for it and appended to it. BCC also is an option for SEND actions at Layer 1; but it does not occur at Layer 3 or higher.

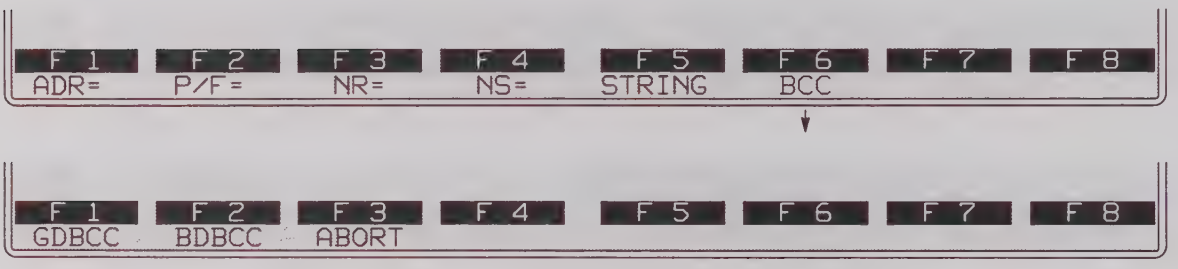


Figure 37-26 Type of BCC is a *SEND* option for frames at Layer 2.

The three softkey selections for BCC are shown in Figure 37-26. A 16–bit CCITT frame check is selected automatically for BOP protocols and cannot be changed or disabled. A bad BCC will be CRC–16 instead of CCITT.

When ABORT is the BCC selection, instead of appending a proper frame check, the transmitter will hold the lead at mark for eight bits (or longer if the transmitter is idling ^F). Inside a frame, seven 1–bits in a row are sufficient to signal an abort.

(B) Give Data

GIVE_DATA is the F2 action on the first rack of action softkeys (refer to Figure 37-19). This action takes the I–field from a received INFO frame and passes it up to Layer 3 along with a DL_DATA IND primitive. (See Figure 34-5 in the section, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data is delivered up to Layer 3 only by one of two actions at Layer 2: GIVE_DATA, or else a DL_DATA IND primitive followed by the data string.

(C) Resend

The RESEND function is mapped to F1 on the second layer of action softkeys at Layer 2 for X.25. See Figure 37-27. The first RESEND action will resend the first frame in the window. The window is a queue that buffers INFO frames for retransmission in case one or more transmissions are lost or in error.

The first frame in the window always is the *earliest outstanding* (unacknowledged) frame. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first–frame" position is established. The first RESEND after an acknowledgment always sends the first window frame.

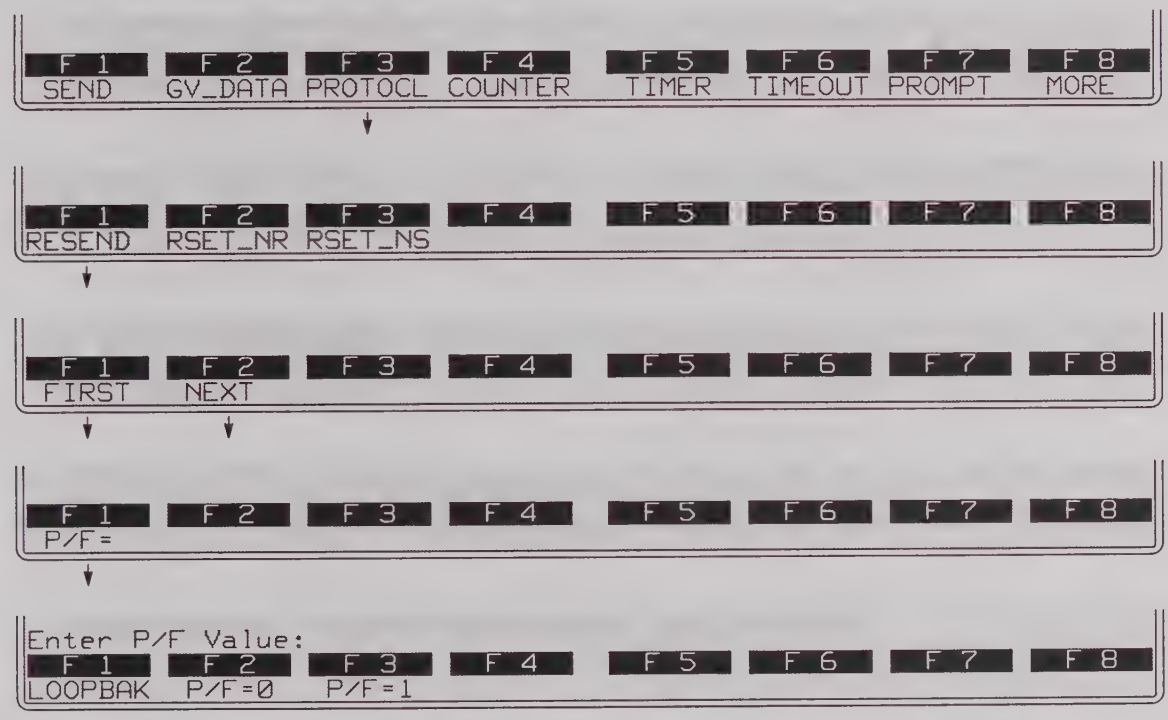


Figure 37-27 The *RESEND* action allows you to recover from sequence errors.

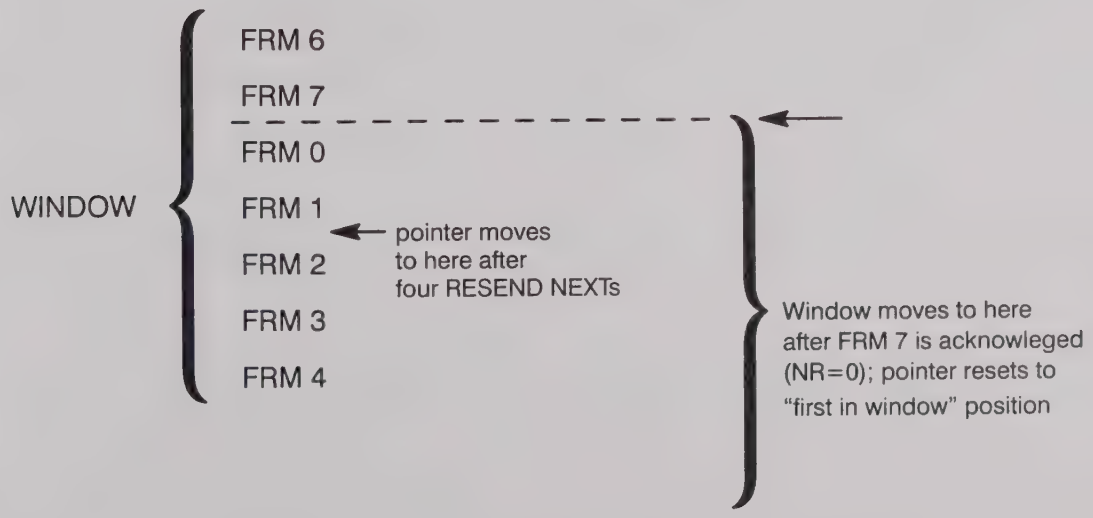


Figure 37-28 Resends cause the pointer to move, while acknowledgments move the pointer *and the entire window.*

The *second and subsequent* RESENDs following an acknowledgment also will send the first window frame, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window frames. Figure 37-28 shows the position of the the resend "pointer" after four consecutive RESEND NEXT actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

The resend–pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

1. *Resend first/next.* RESEND FIRST means that the resend–pointer is reset to the beginning of the window, the first frame in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 37-29.

   The RESEND FIRST action makes it possible for you to resend all the frames in the window one by one, and then *resend them again* if necessary.

2. *P/F=loopback/0/1.* The P/F–bit in the resend–frame can be set to 0 or 1 by this optional action. If PF= LOOPBACK, the bit will echo the last P/F–bit received. (Default is 1 in a RESEND action.)

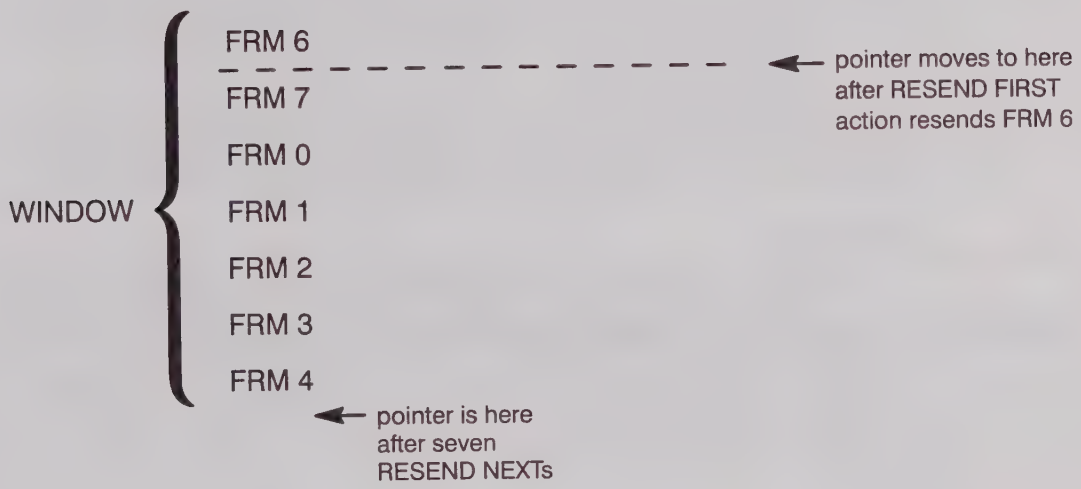


Figure 37-29 *RESEND FIRST* resets the pointer, allowing you to resend the entire window *repeatedly*.

(D) Reset N(R) and Reset N(S)

RESET_NR and RESET_NS are the F2 and F3 actions on the second rack of action softkeys for X.25 Layer 2. (Refer again to Figure 37-19.) The sequence–number fields in I–frames and supervisory frames can be reset by these two Protocol Spreadsheet actions. Sequence numbers are not reset *automatically* during a test by any frame that is sent or received.

RESET_NS also clears the transmit window.

37.6 Display Actions

ENHANCE and SUPPRES pertain to lines of data on the Layer 2 protocol trace (see Section 37.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the X.25 Layer 2 protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F2 on the third rack of action softkeys. Figure 37-30 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

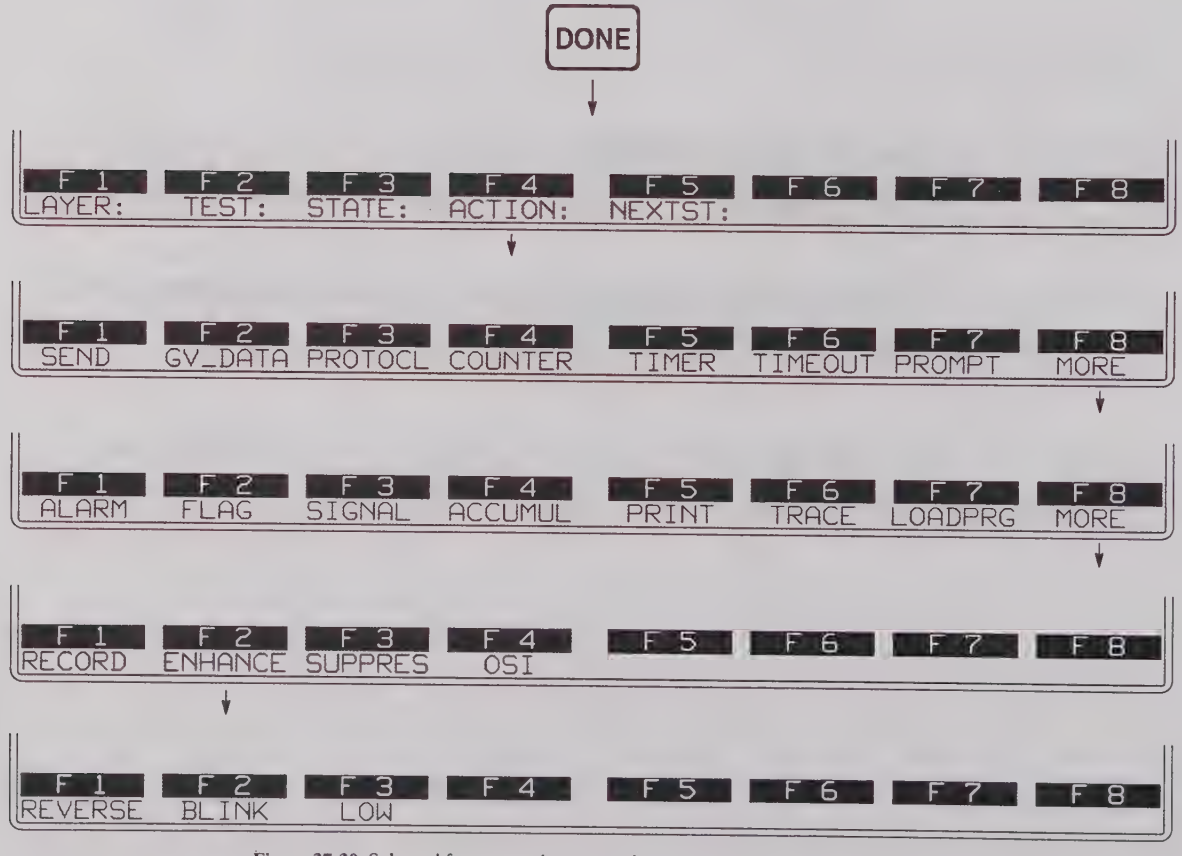


Figure 37-30 Selected frames on the protocol trace may be enhanced or suppressed.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

MON/DISK/FD2 BLK=00838 P 06/16/89 17:06
ASCII/8/NONE/BOP

SRC	ADDR	TYPE	Nr	Ns	P/F	SIZE	TIME	BCC
DCE	01	RR	2		0	0002	1706:03.062	G
DCE	03	INFO	2	0	0	0030	1706:03.179	G
DTE	03	RR	1		0	0002	1706:03.213	G
DCE	03	INFO	2	1	0	0008	1706:03.444	G
DTE	03	RR	2		0	0002	1706:03.485	G
DTE	01	INFO	2	2	0	0009	1706:03.727	G
DCE	01	RR	3		0	0002	1706:03.842	G
DTE	01	INFO	2	3	0	0021	1706:03.910	G
DTE	01	INFO	2	3	1	0021	1706:04.163	G
DCE	01	RR	4		1	0002	1706:04.202	G
DCE	03	INFO	4	2	0	0007	1706:04.384	G
DTE	03	RR	3		0	0002	1706:04.505	G
DCE	03	INFO	4	3	0	0018	1706:04.696	G
DTE	03	RR	4		0	0002	1706:04.741	G
DTE	01	INFO	4	4	0	0007	1706:04.844	G

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8
L2TRACE L3TRACE

Figure 37-31 A retransmitted DTE I–frame has been enhanced.

Figure 37-31 shows one screen of a Layer 2 protocol trace in which DTE INFO frames with the poll bit set to 1—retransmitted DTE INFO frames, in other words—have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE INFO P/F= 1
ACTIONS: ENHANCE REVERSE

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 37-30 shows the softkey path to SUPPRES.

37.7 Automatic Primitives

In a Section 34, Table 34-2 lists the OSI service primitives that are monitored at the boundaries of Layer 2 as trigger conditions, and sent up to Layer 3 or down to Layer 1, as user–entered spreadsheet actions. These primitives are layer–specific rather than

protocol–specific and are not part of the personality package for X.25 Layer 2; but a few of the primitives are set in motion automatically by X.25 Layer 2 spreadsheet actions. These *automatic* primitives can be thought of as part of the Layer 2 actions themselves, and by extension, as part of the X.25 protocol package.

Table 37-2 gives the set of X.25 Layer 2 actions that have action–primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 2, a DL_DATA IND primitive is forwarded to Layer 3, where a DL_DATA IND condition may be waiting to monitor it.

Whenever a SEND or RESEND action is initiated at Layer 2, a PH_DATA REQ primitive is sent downward along with the PH data (the entire frame).

If a SEND or RESEND action is triggered at Layer 2 while the physical connection at Layer 1 is inactive, Layer 2 will sense the absence of a physical connection and delay the PH_DATA REQ. Instead it will send a PH_ACTIVATE REQ primitive. Only when a PH_ACTIVATE CONF has been returned by Layer 1 will Layer 2 release the data and the data primitive.

NOTE: The RS–232 interface does not distinguish active/inactive status at the physical level. This interface returns PH_ACTIVATE CONF automatically whenever it sees PH_ACTIVATE REQ.

Table 37-2
Automatic Primitives Generated at X.25 Layer 2

X.25 Layer 2 Action	Automatic Primitive	To Layer
GIVE_DATA	DL_DATA IND	3
SEND {TYPE}	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1
RESEND	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1

*Sent if Layer 1 shows inactive status. PH_DATA REQ delayed until PH_ACTIVATE CONF returned by Layer 1.

37.8 Programming Example: Converting Protocol Bytes to Hexadecimal

Listed below is a simple, four–trigger test that enhances the "readability" of the raw display of X.25 data. When this test is entered on the trigger menus or anywhere in the Protocol Spreadsheet program at *Layer 1*, frame–level and packet–level protocol bytes on the raw–data display are converted automatically to hexadecimal, while all user data is translated into ASCII characters. Figure 37-32 shows the same data both *before* and *after* the test has been applied.

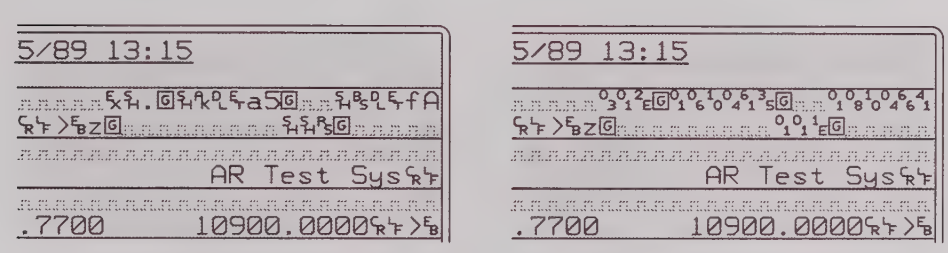


Figure 37-32 In the X.25 data on the right, a simple enhancement program has converted protocol bytes to hexadecimal.

Because it occupies a single state, the protocol–hex test can be entered on the trigger menus; or it may be included in the Protocol Spreadsheet program in this form:

```
LAYER: 1
   TEST: ptcl_hex
      STATE: enhance
         CONDITIONS: DTE STRING "F⊟((XXXXXXX0)) ((0XXXXXXX))☒((XXXXXXX0)) "
         ACTIONS: ENHANCE DTE HEX OFF
         CONDITIONS: DTE GOOD_BCC
         ACTIONS: ENHANCE DTE HEX ON
         CONDITIONS: DCE STRING "F⊟((XXXXXXX0)) ((0XXXXXXX))☒((XXXXXXX0)) "
         ACTIONS: ENHANCE DCE HEX OFF
         CONDITIONS: DCE GOOD_BCC
         ACTIONS: ENHANCE DCE HEX ON
```

In the strings, F ⊟ (FLAG NOT EQUAL FLAG) looks for the beginning of the frame. The first bit mask—((XXXXXXX0))—looks for I–frames. The second bit mask looks for a Q ("*data–qualified*") bit that equals zero. The ☒ entry (DON'T CARE) causes the search string to skip one byte, the LCN byte in the packet header. The third bit mask looks for data packets. Only *unqualified* data packets satisfy the string and turn the hex enhancement *off*. The user data that follows is translated into ASCII.

The test in this form makes two assumptions: the numbering mode is MOD 8 and apart from X.29, no protocol is implemented above Layer 3.

37.9 Programming Example: A Simple "Automatic" Layer 2 X.25 Test

Here is a simplified test that makes Layer 2 "automatic" and "transparent" when you are working at Layer 3 or higher.

The test initiates a link–startup sequence when a DL_CONNECT REQ primitive arrives from Layer 3. This primitive will be sent down automatically at Layer 3 as soon as a SEND RESTART or other Layer 3 SEND action is attempted.

Automatic Layer 2 X.25 Test:

```
LAYER: 2

    STATE: init

        CONDITIONS: DL_CONNECT REQ
        ACTIONS: SEND SABM P/F=1
            TIMEOUT t1 RESTART 3.000

        CONDITIONS: RCV UA P/F=1
        ACTIONS: DL_CONNECT CONF
            TIMEOUT t1 STOP
            RESET_NR
            RESET_NS
        NEXT_STATE: info_xfr

        CONDITIONS: TIMEOUT t1
        ACTIONS: SEND SABM P/F=1
            TIMEOUT t1 RESTART 3

        CONDITIONS: RCV SABM
        ACTIONS: DL_CONNECT IND
            TIMEOUT lyr3_resp RESTART 1

        CONDITIONS: DL_CONNECT RESP
        ACTIONS: SEND UA P/F= LOOPBACK
            RESET_NR
            RESET_NS
        NEXT_STATE: info_xfr

        CONDITIONS: TIMEOUT lyr3_resp
        ACTIONS: PROMPT "Layer 3 not responding"

    STATE: info_xfr

        CONDITIONS: DL_DATA REQ
        ACTIONS: SEND INFO '' ((DL_DATA)) ''

        CONDITIONS: RCV INFO
        ACTIONS: SEND RR RESP
            GIVE_DATA

        CONDITIONS: T1_EXPIRED
        ACTIONS: RESEND FIRST P/F= 1

        CONDITIONS: RCV REJ RESP
        NEXT_STATE: xmt_wndw
```

```
STATE: xmt_wndw

   CONDITIONS: ENTER_STATE
   ACTIONS: RESEND FIRST

   CONDITIONS: FRAME_SENT
      MORE_TO_RESEND
   ACTIONS: RESEND NEXT

   CONDITIONS: FRAME_SENT
      NO_MORE_TO_RESEND
   NEXT_STATE: info_xfr
```

The link startup also can be initiated from "below," by a SABM received from the other side of the link. Note that in this Layer 2 program, the UA response to a SABM is made contingent on a Layer 3 program above that acknowledges the link–startup (DL_CONNECT) indication. The link establishment is designed to fail (while displaying the operator prompt, "Layer 3 not responding") unless the following minimum Layer 3 program is included on the Protocol Spreadsheet:

```
LAYER: 3

   STATE:  dl_connect

      CONDITIONS: DL_CONNECT IND
      ACTIONS: DL_CONNECT RESP
```

Here is a fuller Layer 3 program that initiates link setup and also handles the transfer of Restart packets that brings the interface to the Ready (for calls) state.

```
LAYER: 3

   STATE:  dl_connect

      CONDITIONS: ENTER_STATE
      ACTIONS: DL_CONNECT REQ

      CONDITIONS: DL_CONNECT CONF
      NEXT_STATE: packet_level_ready

      CONDITIONS: DL_CONNECT IND
      ACTIONS: DL_CONNECT RESP
      NEXT_STATE: packet_level_ready


   STATE: packet_level_ready

      CONDITIONS: ENTER_STATE
      ACTIONS: SEND RESTART

      CONDITIONS: RCV RESTART
      ACTIONS: RESET_PR_PS
      NEXT_STATE: ready_to_call

      CONDITIONS: RCV RESTART_CONF
      ACTIONS: RESET_PR_PS
      NEXT_STATE: ready_to_call

   STATE: ready_to_call
```

In **info_xfr** state, the test receives DL_DATA from Layer 3 and *sends* it down to Layer 2. It *gives* DL_DATA up to Layer 3. Recovery actions are taken, as follows: when T1 expires, the test resends the first frame in the window. When a REJ frame is received, the test moves to **xmt_wndw** state and resends the entire window before returning to **info_xfr** state.

38 X.25 Layer 3

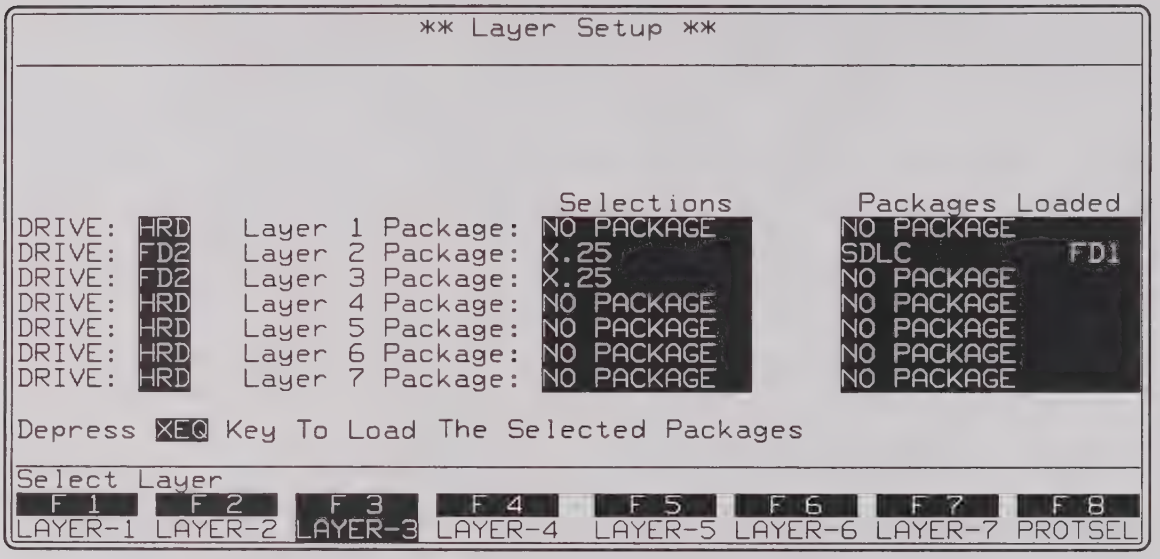


Figure 38-1 The X.25 personality package for Layer 3 is loaded from the Layer Setup screen.

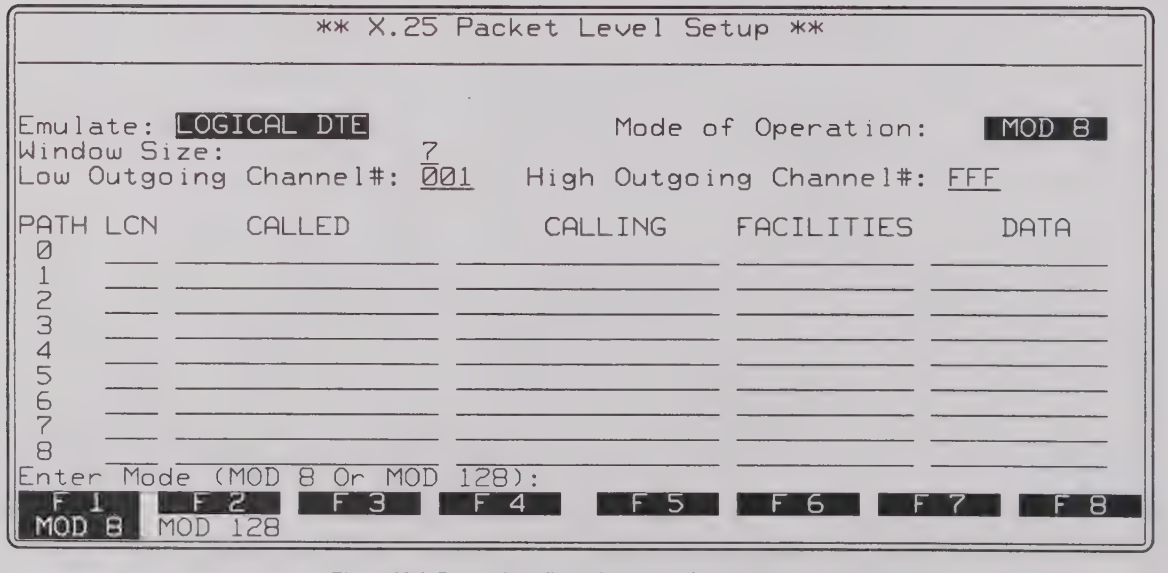


Figure 38-2 Protocol configuration screen for X.25 Layer 3.

38 X.25 Layer 3

Layer 3 X.25 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 38-1 shows the Layer Setup screen configured to load in the Layer 2 and Layer 3 X.25 packages from floppy−disk, Drive 2. Refer to Section 8 for details on operating the Layer Setup screen.

The Layer 3 X.25 package consists of the following:

- A special X.25 Packet Level Setup screen (shown in Figure 38-2) that controls certain parameters when the unit is tracing or emulating X.25.
- A protocol trace (illustrated in Figure 38-4) that distills from X.25 data the packet−level events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 3 on the Protocol Spreadsheet that facilitate X.25 programming. Figure 38-9 shows the softkey path to the first rack of condition softkeys when the X.25 package is loaded in at Layer 3.

38.1 Packet−Level Setup

The X.25 Packet Level Setup screen must be configured correctly for an accurate trace display and for proper emulation.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the X.25 selection at Layer 3: **X.25** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up the prompt to **Select Protocol Configuration Screen**. Then press F3 (LAYER−3) to call up the X.25 Packet Level Setup screen.

The five parameter fields on this screen as well as the path−data entry fields are shown in Figure 38-2. **Emulate, Window Size, Low Outgoing Channel #, High Outgoing Channel #**, and the entire path−data area apply only to interactive (emulate) tests. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) Emulate Logical DTE/DCE

There are two selections in the **Emulate** field on the X.25 Packet Level Setup screen, LOGICAL DTE and LOGICAL DCE. The entry in this field determines the order of assignment when LCNs are assigned dynamically to call requests during interactive testing.

Configured as a logical DTE, the INTERVIEW assigns LCNs in a descending sequence beginning with the High Outgoing Channel #. See Section (D), below. Configured as a logical DCE, the INTERVIEW assigns LCNs in ascending sequence beginning with the Low Outgoing Channel #.

(B) Mode of Operation

The **Mode of Operation** field selects the mode of numbering DATA and supervisory packets. There are two options, MOD 8 and MOD 128.

MOD 8 uses sequence numbers 0–7. MOD 128 adds an extra byte to the control field in DATA, RR, RNR, and REJ packets. See Figure 38-6. This extra byte allows sequence numbers in a range of 0–127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate X.25 Layer 3 trace.

(C) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged data packets that Layer 3 will buffer for retransmission. When the limit is reached, any further data packets that are named in SEND actions triggered at Layer 3 will be passed to Layer 2 but not buffered for resending.

According to CCITT Recommendation X.25, the standard window size is 2. This means that two packets can be outstanding (unacknowledged) at a time.

The window is a queue that buffers packets for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) packet in the window. Successive RESENDs will send successive packets until there are no more packets to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

(D) Low Outgoing Channel#/High Outgoing Channel#

Logical Channel Numbers (LCNs) may be assigned dynamically on a per–call basis; or they may be reserved during Program mode for a particular call destination (or "path") by means of an entry in the **LCN** column on the X.25 Packet Level Setup screen.

If the **LCN** column on the Packet Level Setup screen is left blank with respect to a particular call destination ("path"), an LCN will be assigned dynamically each time the INTERVIEW initiates a call on that network path, as follows:

If the INTERVIEW has been configured as a logical DCE on the Layer 3 Protocol Configuration screen, it will select the Low Outgoing Channel number for the first call to the DTE. Assuming that the first call still is in session, the next call initiated by the INTERVIEW will have the next higher LCN, and so forth, to the upper limit for LCNs established in the **High Outgoing Channel#** menu field.

If the INTERVIEW has been configured as a logical DTE, the first call initiated by the INTERVIEW will receive the High Outgoing Channel number. If the first call remains in session, the second call will have the second–highest LCN, and so forth, to the lower limit for LCNs set in the **Low Outgoing Channel#** field.

(E) Path

"Path" is the route of a Call Request packet through the X.25 network, on the way to its destination address. Call Request packets establish the path of a call. When data packets enter the network, they carry the same LCN as the call packet. They use the LCN to identify themselves at each network node, where they are routed along the path already established for the call.

Packets that are programmed to be "sent" on the Protocol Spreadsheet must indicate their destination—that is, they must declare the call that they belong to. In the network, they will use the LCN to identify their call. But the LCN cannot be used for identification on the INTERVIEW's Protocol Spreadsheet, since LCNs usually are assigned dynamically at the time of the call—in which case they cannot be preprogrammed.

Instead, packets on the spreadsheet are provided with a "path" number that ties them, on the X.25 Packet Level Setup screen, to a particular set of Call Request parameter values. Here is an example of how the procedure works:

On the Packet Level Setup screen (see Figure 38-3), the following entry is made in the **CALLED** column for Path #0: 300170345678. On the Protocol Spreadsheet, this Actions entry is made: SEND CALL PATH= 0.

When the SEND action is triggered, a Call Request packet will be formed with the Called address given for Path #0 on the Packet Level Setup screen: 300170345678. An LCN will be assigned to this call packet dynamically, according to the rules for low and high outgoing channel numbers outlined in Section 38.1(D), above. As long as the call is active, data packets and other packets sent on this *path*—for example, SEND DATA PATH= 0—will carry the same LCN.

(F) LCN

Normally the LCN field for a particular call (or "path") is left blank. During Run mode when the Call Request is created, the LCN is assigned dynamically according to the rules for low and high outgoing channel numbers described above. For the duration of the call, data packets and other packets are constructed with the same LCN.

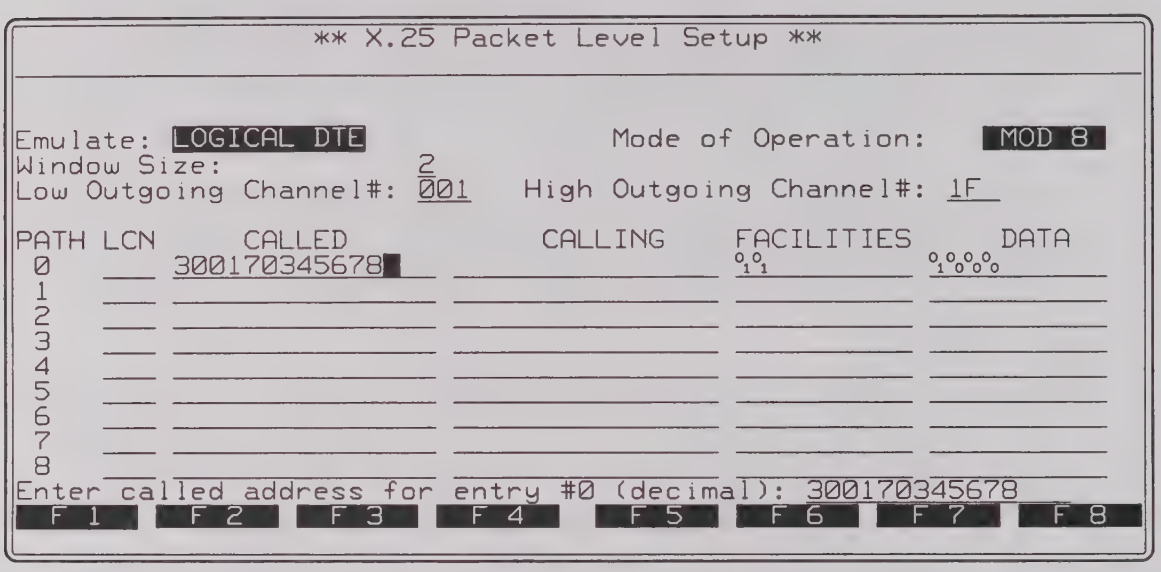


Figure 38-3 Packet–level setup screen with sample data.

An LCN also can be predefined by the user. This designation is made in the **LCN** column of the Packet Level Setup screen, not on the Protocol Spreadsheet. The SEND action on the spreadsheet still references only the path number. On the Packet Level Setup screen, that path number is correlated to the LCN that the user enters on the same row.

The **LCN** field is referred to as a "hex" field. This means that each column (character space) in the data–entry field will equate to one 4–bit, hexadecimal digit on the actual data screen. For example, a data screen may show the two–character sequence $^{0}_{1}{}^{2}_{3}$, where the second, third, and fourth digits represent the LCN. The LCN entry on on the Packet Level Setup screen was 123. Or the character data may show this sequence: $^{0}_{0}{}^{1}_{E}$. The LCN on the setup screen was 1E (also 01E).

(G) CALLED

Enter the called address in this field. Addresses are considered "decimal" entries. This means that each column or character space in the data–entry field will equate to a 4–bit, binary–coded decimal (BCD) digit on the actual data screen. Use the numbered keys to make this entry—do not use the HEX key to turn on the hex function. A sample address entry is shown in Figure 38-3.

(H) CALLING

Enter the calling address in this field. (Logical DTEs often omit this address: the network knows the addresses of dedicated lines coming into the node and may not require them.)

Addresses are considered "decimal" entries. This means that each column or character space in the data–entry field will equate to a 4–bit, binary–coded decimal (BCD) digit on the actual data screen. Use the numbered keys to make this entry—do not use the [HEX] key to turn on the hex function. A sample address entry is shown in Figure 38-3.

Do not make any allowance in this field for the Address Length byte (see Figure 38-6). That byte is provided automatically.

(I) FACILITIES

Enter the entire Facilities field as it will appear in the Call Request packet. *Omit the Facilities Length byte*; that is handled automatically.

The **FACILITIES** field is referred to as a "character" field. This means that characters—including hexadecimal characters—in the data–entry field will equate one–for–one with the characters on the actual data screen.

The Facilities entry in Figure 38-3 will be transmitted in a Call Request packet exactly as it appears in this field. With a Facilities Length byte preceding it, it will look like this on the data display: $^{0}_{2}{}^{0}_{1}{}^{0}_{1}$.

(J) DATA

Enter the Data field as it will appear in the Call Request packet. If you want a Data field that is longer than the ten character spaces in the **DATA** field, you can append a *string* to your Call packet on the Protocol Spreadsheet.

The Data field on the Packet Level Setup screen is a "character" field. This means that the Data entry in Figure 38-3 will be transmitted exactly as it appears in this field. (As in any transmit string on an INTERVIEW screen, in normal bit order the *rightmost* bit in the *leftmost* byte will be the first bit transmitted.)

Based on the entries for Path #0 in Figure 38-3, the INTERVIEW will send the following Call Request *packet* in a SEND CALL PATH= 0 action. Frame–level bytes are not shown. Assume an LCN of 01E.

hex:

$^{1}_{0}{}^{1}_{E}{}^{0}_{8}{}^{0}_{C}{}^{3}_{0}{}^{0}_{1}{}^{7}_{0}{}^{3}_{4}{}^{5}_{6}{}^{7}_{8}{}^{0}_{2}{}^{0}_{1}{}^{0}_{1}{}^{0}_{0}{}^{0}_{0}{}^{0}_{0}$

ASCII:

38.2 Protocol Trace

The Layer 3 X.25 package includes an automatic packet–trace display that summarizes packet–level activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

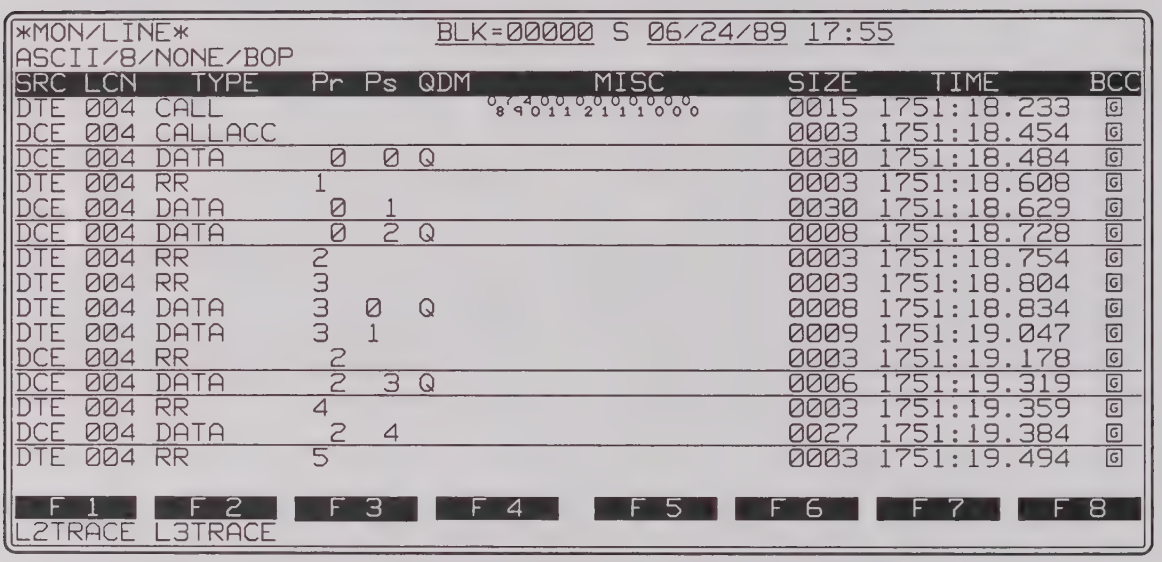


Figure 38-4 Each horizontal row on the trace display represents a packet.

While the unit is in Run mode, press the softkey for PROTOCL (F2 on the primary rack of display–mode softkeys) and then the softkey for L3TRACE (F2)to bring the protocol trace for X.25 Layer 3 to the screen. Figure 38-4 is an example of this trace display. Each horizontal row in the trace represents a packet.

(A) The Protocol Trace in Freeze Mode

Press FREEZE to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing ROLL FWD or ↓ moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing ROLL BACK or ↑ moves the viewing window up to add a line of older data to the top of the screen.

Depression of the NEXT PAGE key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of PREV PAGE adds 15 lines of older frames.

The packet displayed on the top line of frozen trace–data will appear in the first frame in the raw–data or data–plus–leads display. To view the character data that

generated a particular line in the trace display, use [ROLL FWD] or [ROLL BACK] (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys.

For example, the Call Request packet traced on the top line of the display in Figure 38-4 also is contained in the frame at the *top left* of the *data* display in Figure 38-5. This correlation is automatic.

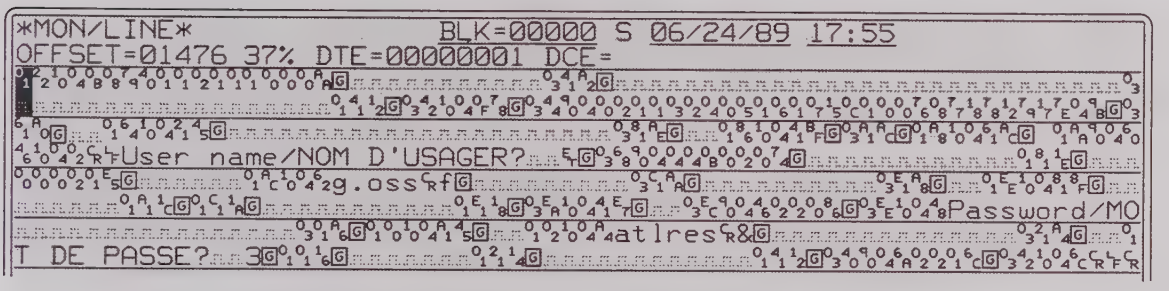


Figure 38-5 Data–display of Protocol Trace shown in Figure 38-4.

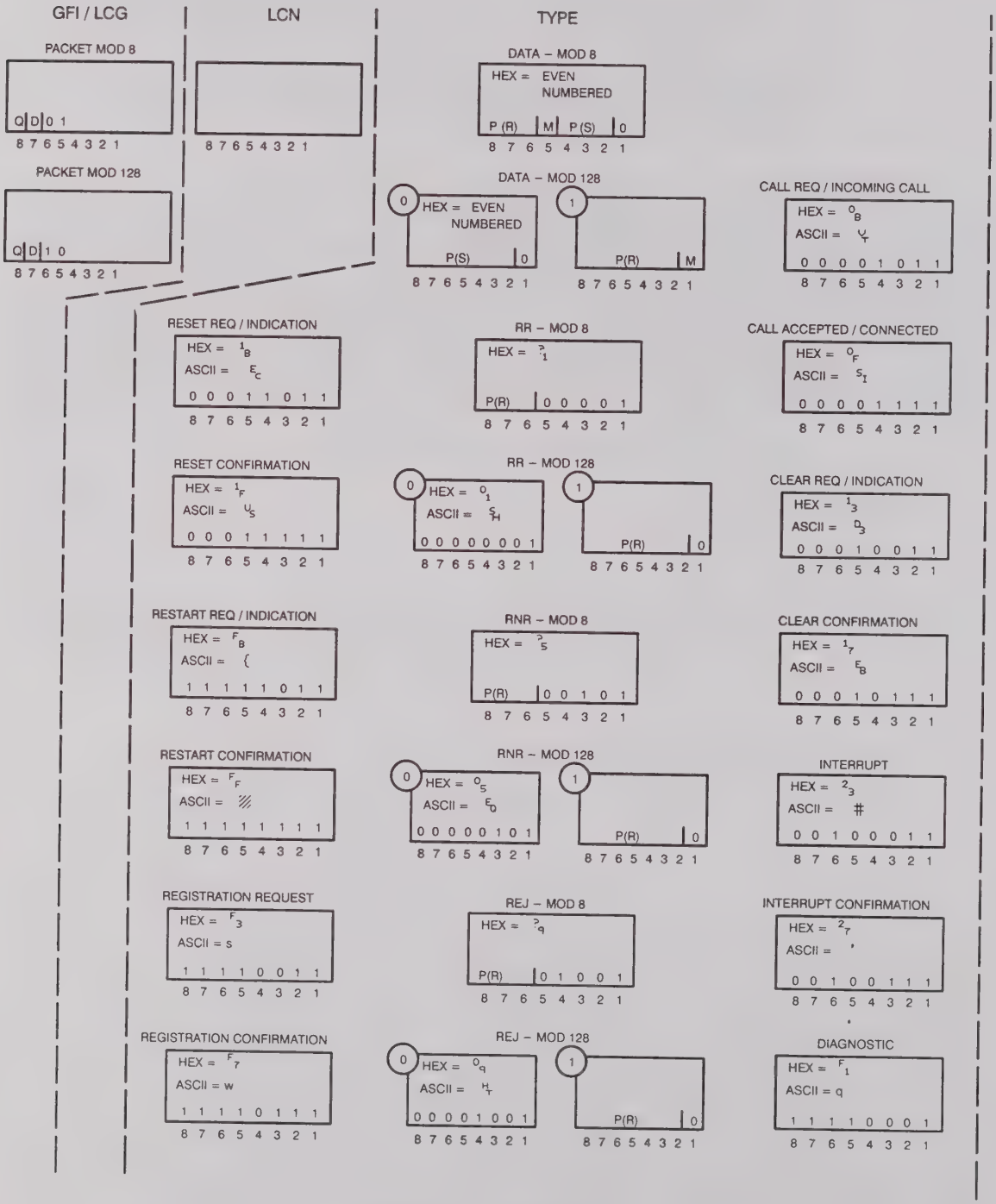
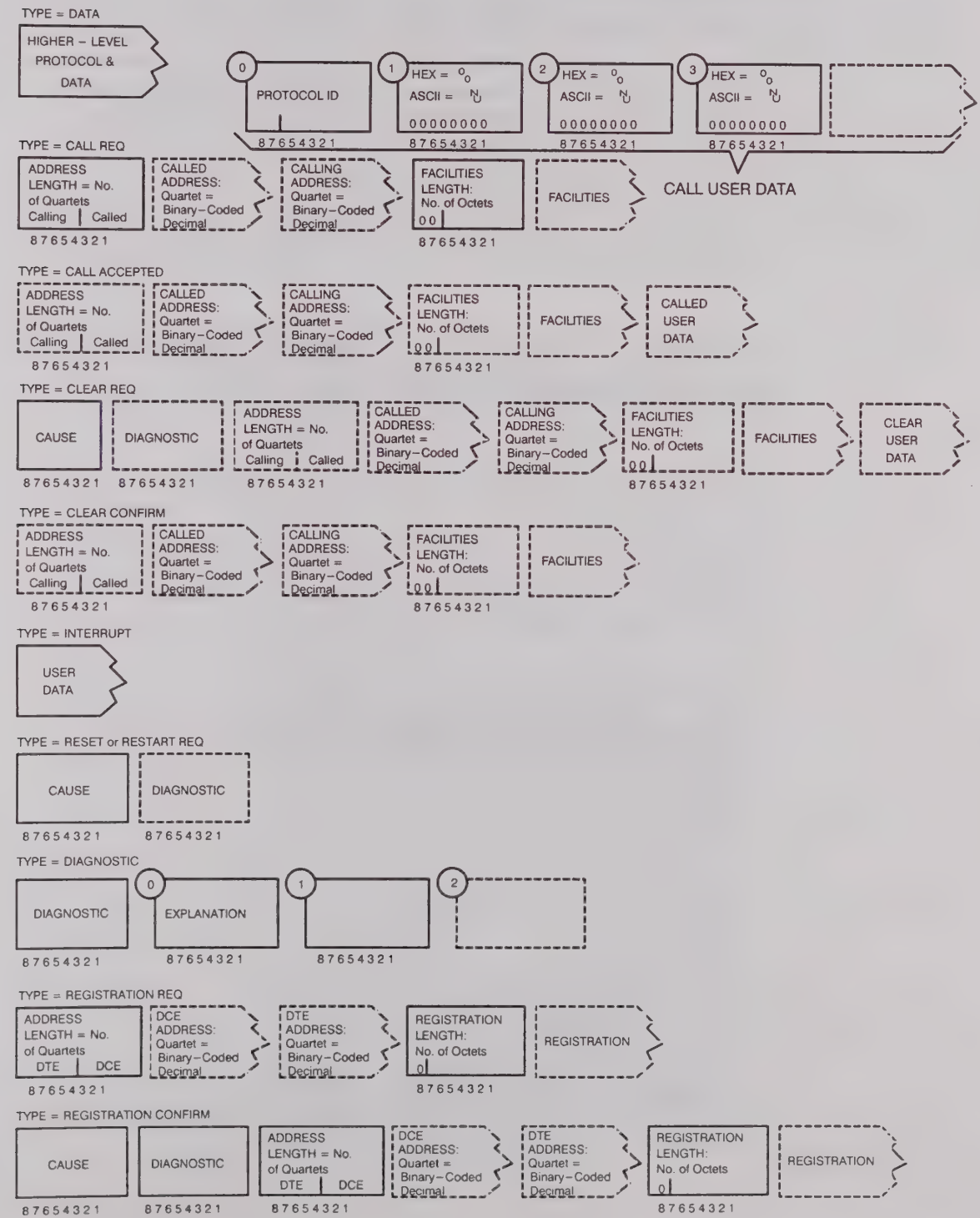


Figure 38-6 X.25 packet fields.

DATA


TYPE = DATA
HIGHER – LEVEL PROTOCOL & DATA
0
PROTOCOL ID
1
HEX = 0
ASCII = N
00000000
2
HEX = 0
ASCII = N
00000000
3
HEX = 0
ASCII = N
00000000
87654321
CALL USER DATA
TYPE = CALL REQ
ADDRESS LENGTH = No. of Quartets
Calling | Called
CALLED ADDRESS: Quartet = Binary–Coded Decimal
CALLING ADDRESS: Quartet = Binary–Coded Decimal
FACILITIES LENGTH: No. of Octets
00
FACILITIES
TYPE = CALL ACCEPTED
CALLED USER DATA
TYPE = CLEAR REQ
CAUSE
DIAGNOSTIC
CLEAR USER DATA
TYPE = CLEAR CONFIRM
TYPE = INTERRUPT
USER DATA
TYPE = RESET or RESTART REQ
TYPE = DIAGNOSTIC
EXPLANATION
TYPE = REGISTRATION REQ
DCE ADDRESS: Quartet = Binary–Coded Decimal
DTE ADDRESS: Quartet = Binary–Coded Decimal
REGISTRATION LENGTH: No. of Octets
REGISTRATION
TYPE = REGISTRATION CONFIRM

(B) Trace Columns

The columns in the protocol trace for Layer 3 X.25 are explained below.

1. *Source*. The **SRC** column identifies the lead on which the packet was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the packet, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

   Just as on the data display, RD data is underlined.

2. *LCN*. The LCN for each packet is given in this three–column "hex" field. Each column displays a hexadecimal *digit* (0 through F) that represents four bits out of the 12–bit LCN.

3. *Type*. The mnemonic (abbreviated) names for seventeen packet types as they appear in the **TYPE** column of the protocol trace are shown in Figure 38-6 under "TYPE." If a Type octet does not fit any of the patterns in the figure, the packet it listed in the **TYPE** column as UNKWN followed by the hexadecimal value of the type byte: UNKWN=03.

   If the number of bytes in the packet is below the required minimum, the packet is posted as INVALID.

4. *P(R) and P(S)*. One column on the packet–level trace is devoted to P(R) values, and one column to P(S). The packet types that include P(R) or P(S) fields in their control fields are indicated in Figure 38-6. P(R) and P(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

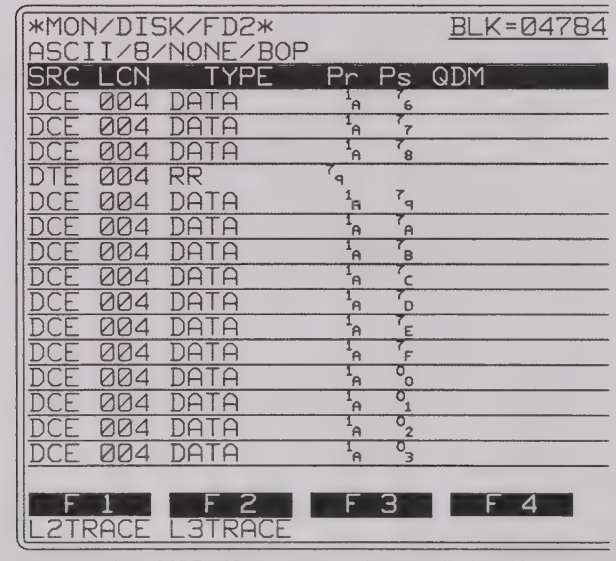


Figure 38-7 MOD 128 sequence numbers are displayed in two–digit hexadecimal characters.

P(R) and P(S) values are presented in decimal format in modulo−8 traces. Each column displays a single digit that represents a 3−bit binary value (0 through 7). For modulo 128, the values % to 7F are given in "character" format, where the columns contain a two−digit hexadecimal character (see Figure 38-7).

Note that the **Pr** and **Ps** columns on the trace are staggered to suggest four columns. The two outside columns, comprised of the DTE's P(R) value and the DCE's P(S) value, form a numbering sequence for DCE data packets. The arrows in Figure 38-8 indicate the sequence: the DCE sends packet 4, the DTE acknowledges 4 by returning P(R) 5; the DCE sends 5, the DTE expects 6; the DCE sends 6, the DTE expects 7; the DCE sends 7, the DTE expects 0; the DCE sends 0.

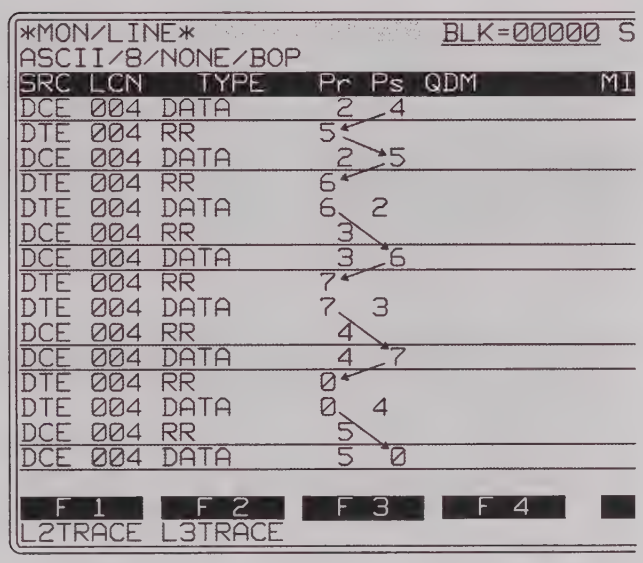


Figure 38-8 *Pr* and *Ps* columns are staggered, with the *outside* columns representing the sequence of *DCE* numbered data packets.

The two inside columns reveal a similar pattern for DTE data packets (and DCE acknowledgements).

5. *Q, D, and M.* **QDM** is a three−column field. If the Q (data−*q*ualified) bit is set in a data packet, a **Q** will appear in the **Q** column for that row. See Figure 38-4 for examples of this letter−Q display. The position of the Q−bit in the first packet byte is indicated at the top left of Figure 38-6.

   When the D (*d*elivery) bit is set in a Call, Call Accept/Connect, or Data packet, the letter **D** appears in the **D** column. The position of the D−bit in the first packet byte is indicated in Figure 38-6.

When the M (*m*ore) bit is set in a data packet, the letter M appears in the **M** column on the trace display. The position of the M–bit in the Type field of data packets is shown in Figure 38-6.

6. *Misc*. The **MISC** field presents up to 16 bytes of character data (decoded in hex) for all packet types *other than data packets* that contain data beyond the Type octet. All such packets and their "miscellaneous" fields are indicated on the right half of Figure 38-6.

   Twelve bytes of "miscellaneous" data were expanded for the Call packet in the trace in Figure 38-4. The data in this example includes the address–length byte, four called–address bytes, the facilities–length byte, two facilities bytes, and four bytes of call–user data.

7. *Size*. The number of bytes in each packet is given in this field in four decimal digits.

8. *Time*. The time of the arrival of the *end of the frame containing the packet* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

   If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

9. *Frame checking*. An X.25 frame ends as soon as a $^{7}\varepsilon$ flag or seven 1–bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

   A for abort is posted to the trace row when a frame is ended by seven 1–bits.

38.3 Monitor Conditions

When the Layer 3 X.25 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 3. See Figure 38-9.

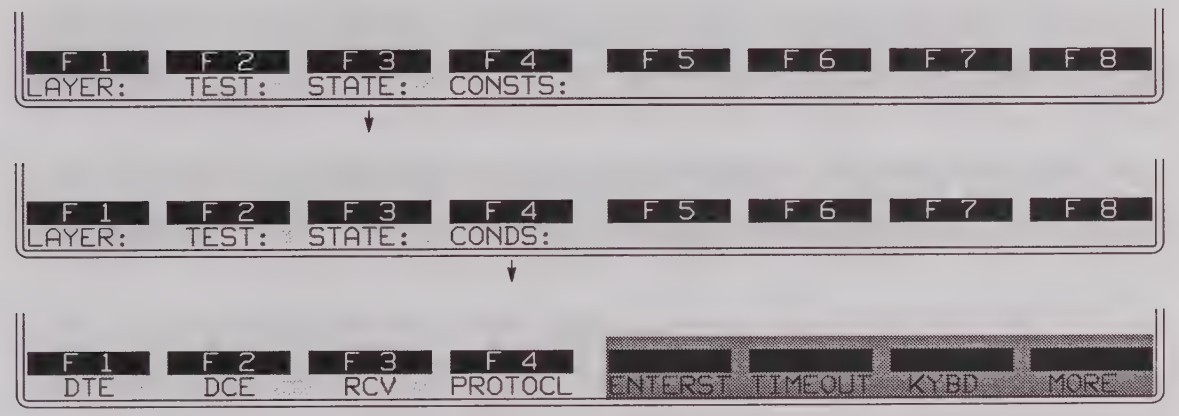


Figure 38-9 Unlike *RCV* conditions, the softkeys for *DTE* and *DCE* are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represents *types* of packets: DATA, RR, RNR, REJ, CALL, and so forth.

(A) Packet Type

The softkeys for data, supervisory, unnumbered, and "other" packets are illustrated in Figure 38-10.

Press a softkey to write one of these packet types to the Layer 3 spreadsheet. DTE or DCE followed by a packet–type mnemonic—DTE DATA, for example, or DCE CLEAR—is a complete condition and will come true if a matching packet is monitored. An LCN condition may be added to the simple packet mnemonic, but it is optional. Other optional conditions that may apply are Q–bit value, D–bit value, M–bit value, cause code, and diagnostic code.

> **NOTE:** A packet–type condition will not come true with respect to a packet that is inside a frame with a bad frame check, or inside an aborted frame.

1. *Data packets*. Data packets differ for MOD 8 and MOD 128 numbering schemes. (See Figure 38-6.) For spreadsheet conditions to match data packets accurately,

the correct numbering system ("**Mode of Operation**") should be selected on the Packet Level Setup screen.

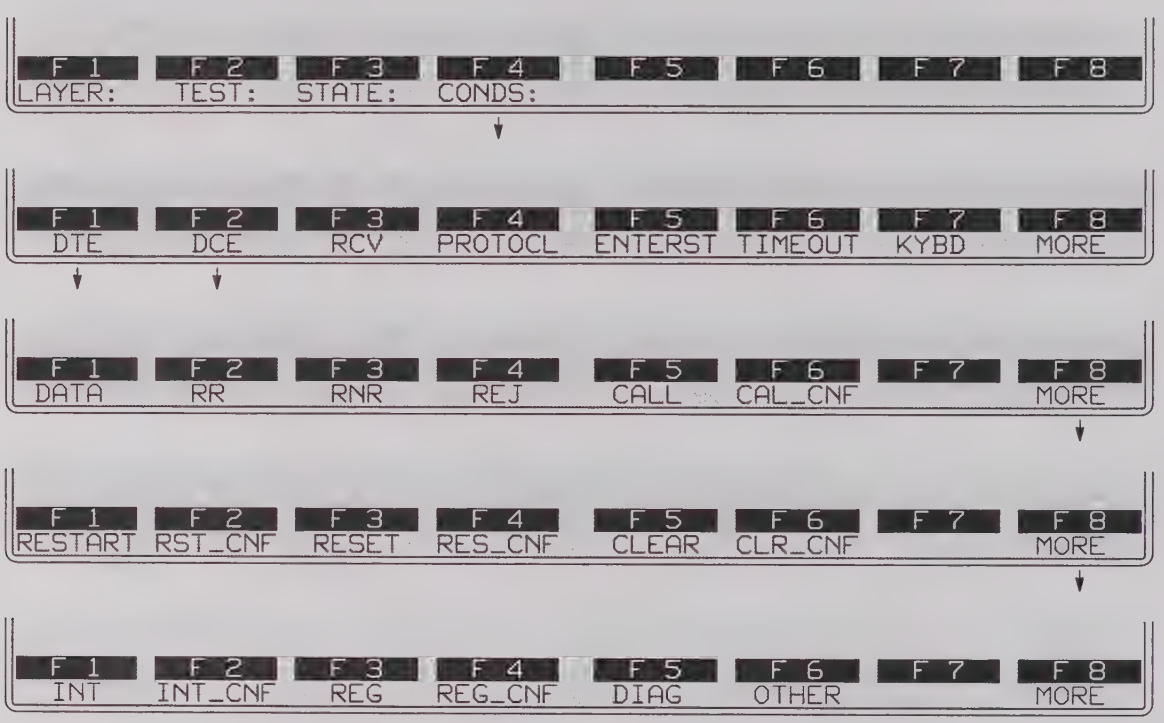


Figure 38-10 Packet types.

When DTE or DCE is monitored for a data packet, a specific LCN *may* be specified in the spreadsheet condition. A specific value for the Q−, D−, or M−bit also may be indicated in the rack of spreadsheet softkeys just below DATA. (Refer to Figure 38-12.)

2. *Supervisory packets*. The three supervisory−packet types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), and REJect. These packets always contain P(R) fields (see Figure 38-6) and serve mainly to acknowledge or reject data packets.

   Like data packets, supervisory packets are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

   When DTE or DCE is monitored for a supervisory packet, a specific LCN may be specified in the spreadsheet condition. See Figure 38-11.

3. *Unnumbered packets*. Unnumbered packets generally assist in call setup, call management, call clearing, and subscription services.

The thirteen unnumbered packet types are laid out consecutively from CALL to DIAG on the softkey racks in Figure 38-10. Because these packets lack P(R) and P(S) sequence numbers, they are constructed identically for MOD 8 and MOD 128.

All unnumbered–packet conditions may be made specific to a particular LCN. Call and Call Confirm conditions may specify a D–bit value (Figure 38-13). Restart, Reset, Clear, and Registration Confirm conditions may optionally test for causes and diagnostic codes (see Figure 38-14, Figure 38-15, and Figure 38-16). Diagnostic packets (F5 on on the bottom rack of softkeys in Figure 38-10) also may specify a diagnostic code.

4. *Other packets*. Any packet type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. (See F6 in the bottom rack of softkeys in Figure 38-10.) Then enter the hex byte in the form of two alphanumerics. Here, for example, is a Clear Request entered as a hexadecimal:

```
CONDITIONS: DCE OTHER 13
```

(B) LCN

All DTE and DCE conditions that name a packet type may specify one particular LCN (logical channel number) as an added condition. For example, a spreadsheet condition may be satisfied by *any* DTE Clear Request packet:

```
CONDITIONS: DTE CLEAR
```

Or it may be satisfied by a DTE Clear Request packet only if it carries an LCN of, say, 005:

```
CONDITIONS: DTE CLEAR LCN= 005
```

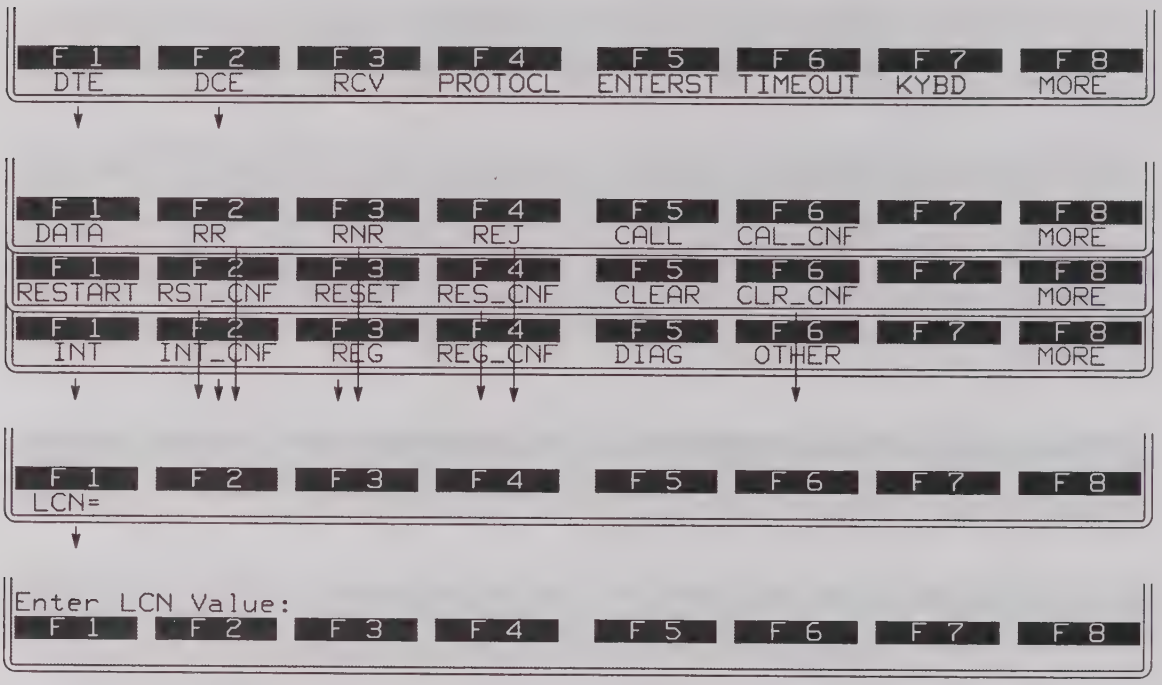


Figure 38-11 LCN is the Condition option available for these nine packet types.

Figure 38-11 indicates the packet types that offer LCN as their only Condition option.

Enter the LCN as one, two, or three hexadecimal digits. Type each digit as an alphanumeric in the range *0−9* and *A−F* (or *a−f*): *do not use the* [HEX] key. Each digit will represent four bits of the 12−bit LCN. A single−digit or two−digit entry will represent the low−order bits, with the high−order bits zero−filled. Thus LCN= 005 is the same entry as LCN= 05 or LCN= 5.

(C) Q, D, and M Bits

Q−, D−, and M−bit values of 0 or 1 may be specified in Layer 3 conditions that search for DATA packets. See Figure 38-12.

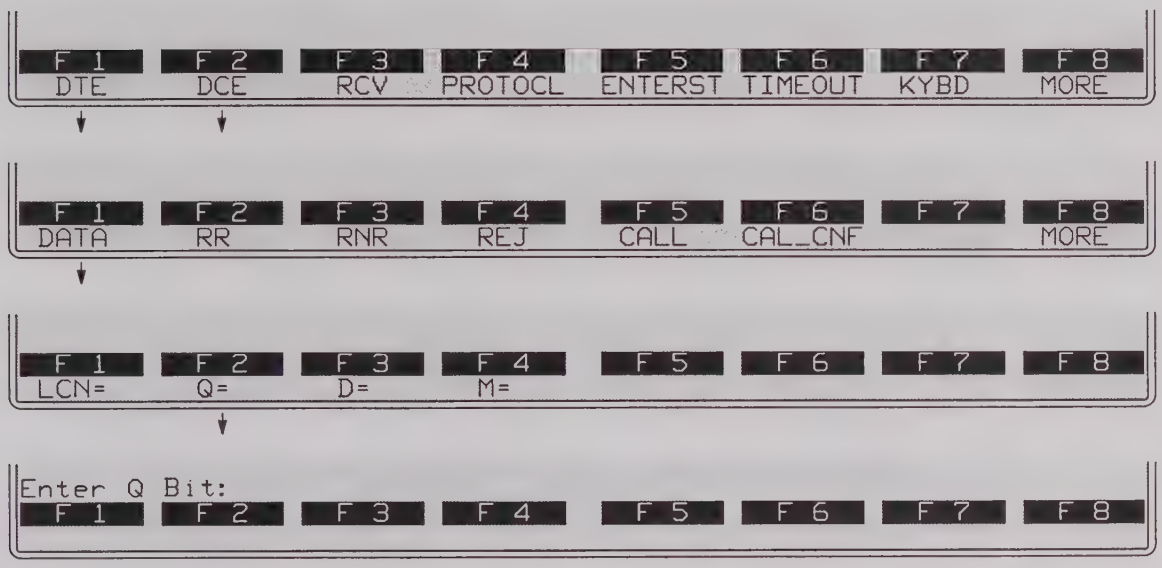


Figure 38-12 When data packets are monitored, Q, D, and M values of 1 (along with LCN values) *may* be specified.

A D−bit value also may be specified for Call and Call Confirm packets: see Figure 38-13.

The positions of the Q−, D−, and M−bits in the packet header are illustrated in Figure 38-6.

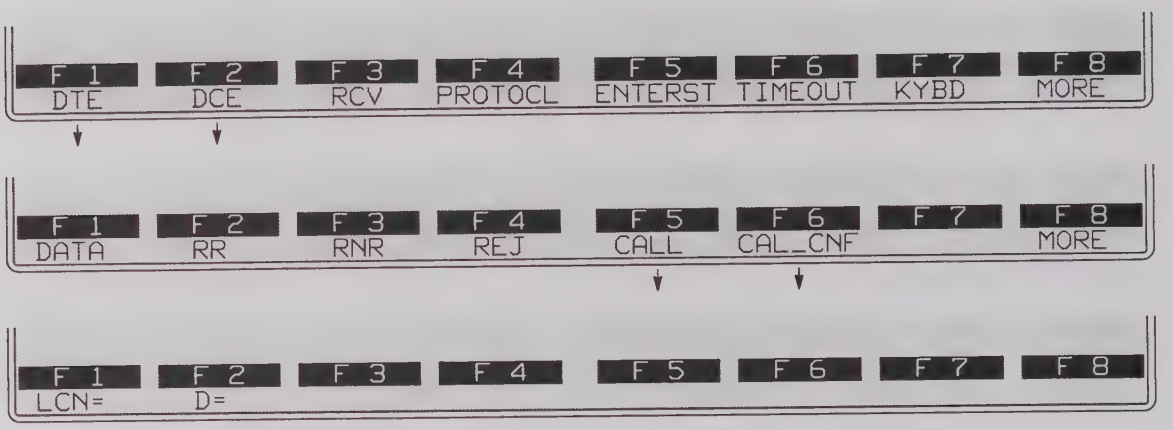


Figure 38-13 Conditions that look for Call and Call Confirm packets also can test for LCN and D−bit values.

(D) Cause and Diagnostic Value

Conditions that look for Restart, Reset, Clear, and Registration Confirm packets may be refined further to test for a particular cause code and/or diagnostic code.

1. *Cause*. The names of causes as well as their hexadecimal values are indicated on the softkey−prompt line near the bottom of the Protocol Spreadsheet screen. To specify a particular cause, the user does not have to memorize cause codes or consult a table. The user simply presses [F2], ROLL, and repeats the keystroke to cycle through the list of cause names for a given packet type. Figure 38-14 shows the cycle of causes that pertain to Restart packets. The user presses [F1], SELECT, when the right cause has "rolled" up onto the prompt line. The SELECT softkey writes the current cause onto the spreadsheet screen.

   Here is an example of a cause−code entry on the Protocol Spreadsheet:

   ```
   CONDITIONS: DCE CLEAR CAUSE= NOT_OBTAINABLE
   ```

   Causes also may be entered into the spreadsheet test as two hexadecimal digits, as in this example:

   ```
   CONDITIONS: DCE CLEAR CAUSE= 0D
   ```

   Notice that each digit is an alphanumeric in the range *0−9* and *A−F* (or *a−f*): *do not use the* [HEX] *key*.

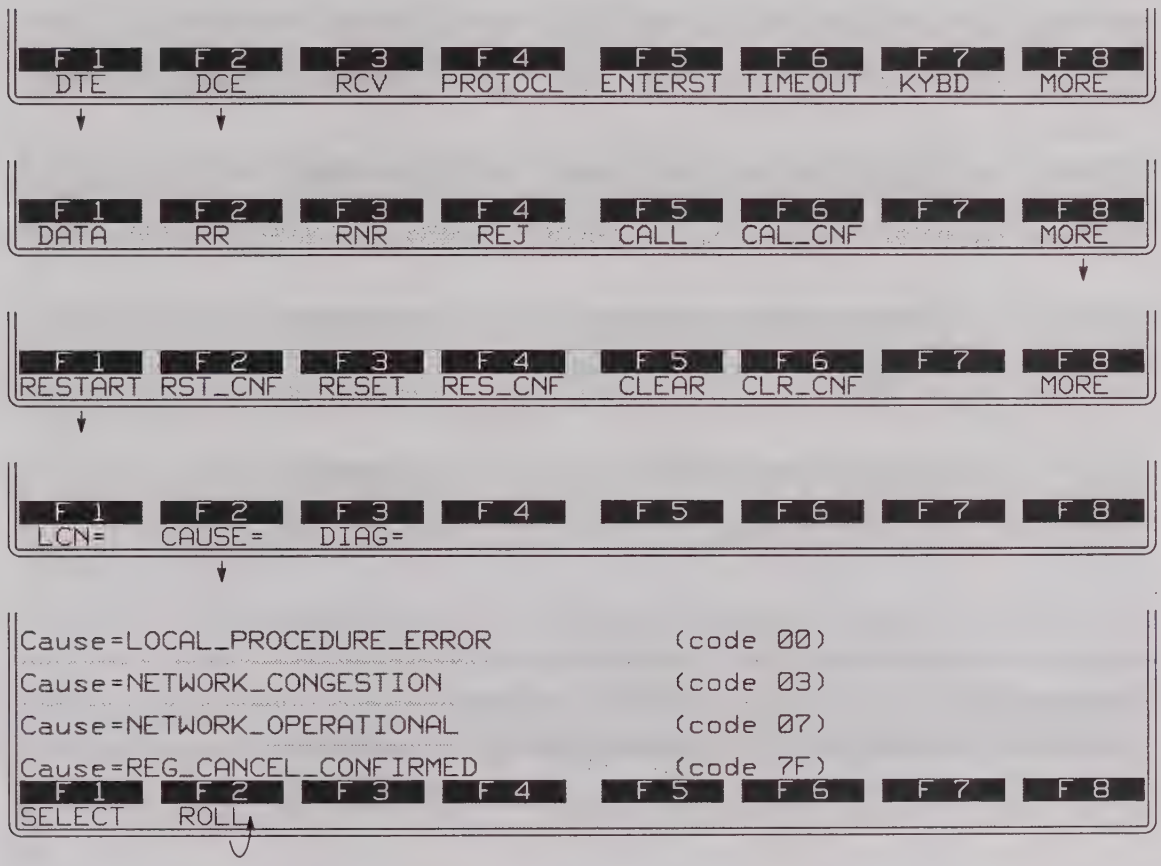


Figure 38-14 A Restart packet may be tested for one of these four causes.

(a) Reset causes

```
Cause=DTE_ORIGINATED                    (code 00)
Cause=OUT_OF_ORDER                      (code 01)
Cause=REMOTE_PROCEDURE_ERROR            (code 03)
Cause=LOCAL_PROCEDURE_ERROR             (code 05)
Cause=NETWORK_CONGESTION                (code 07)
Cause=REMOTE_DTE_OPERATIONAL            (code 09)
Cause=NETWORK_OPERATIONAL               (code 0F)
Cause=INCOMPATIBLE_DESTINATION          (code 11)
Cause=NETWORK_OUT_OF_ORDER              (code 1D)
 F1    F2    F3    F4    F5    F6    F7    F8
SELECT   ROLL
```

(b) Clear causes

```
Cause=DTE_ORIGINATED                    (code 00)
Cause=NUMBER_BUSY                       (code 01)
Cause=OUT_OF_ORDER                      (code 09)
Cause=REMOTE_PROCEDURE_ERROR            (code 11)
Cause=REVERSE_CHARGE_NOT_ACCEPTED       (code 19)
Cause=INCOMPATIBLE_DESTINATION          (code 21)
Cause=FAST_SELECT_NOT_ACCEPTED          (code 29)
Cause=SHIP_ABSENT                       (code 39)
Cause=INVALID_FACILITY_REQUEST          (code 03)
Cause=ACCESS_BARRED                     (code 0B)
Cause=LOCAL_PROCEDURE_ERROR             (code 13)
Cause=NETWORK_CONGESTION                (code 05)
Cause=NOT_OBTAINABLE                    (code 0D)
Cause=RPOA_OUT_OF_ORDER                 (code 15)
 F1    F2    F3    F4    F5    F6    F7    F8
SELECT   ROLL
```

(c) Registration Confirmation causes

```
Cause=REG_CANCEL_CONFIRMED              (code 7F)
Cause=INVALID_FACILITY_REQUEST          (code 03)
Cause=LOCAL_PROCEDURE_ERROR             (code 13)
Cause=NETWORK_OPERATIONAL               (code 05)
 F1    F2    F3    F4    F5    F6    F7    F8
SELECT   ROLL
```

Figure 38-15 The various causes available for *(a)* Reset, *(b)* Clear, and *(c)* Registration Confirm packets.

Figure 38-16 These four packet types (plus *Diagnostic*) allow you to enter a diagnostic–code value as a spreadsheet condition.

2. *Diagnostic code*. Diagnostic–code values are optional conditions for the following packet types: Restart, Reset, Clear, Diagnostic, and Registration Confirm. Figure 38-16 shows the softkey sequences that branch down to the diagnostic–code condition for most of these packet types.

   Enter the diagnostic code as two hexadecimal digits. Type each digit as an alphanumeric in the range *0–9* and *A–F* (or *a–f*): *do not use the* [HEX] *key*. Here is an example of a spreadsheet condition that specifies both a cause and a diagnostic code:

   ```
   DCE RESET CAUSE= LOCAL_PROCEDURE_ERROR DIAGNOSTIC= 01
   ```

38.4 Emulate–Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for **Mode:** EMULATE DTE or EMULATE DCE.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data path for X.25 packet types. RCV conditions operate only in emulate modes, and they check only the data path that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV DATA Q looks the same as DCE DATA Q—there are important differences that are noted below.

1. *Access to the data interface.* The INTERVIEW is a *layered* emulator. The significance of this is that Layer 3 and higher layers have no direct access (in Emulate modes) to the physical layer, Layer 1. In practice this means that a RCV condition at Layer 3 *does not see packets on the line*. It only sees packets that are delivered up from Layer 2 by a *user program* at that layer.

   (Similarly, a SEND action at Layer 3 does not in itself send a packet out onto the line. A SEND action merely delivers the packet to Layer 2 —provided that Layer 2 has indicated its readiness to receive data from above.)

   The following program is not any sort of complete Layer 2 emulation. It is the *minimum* program that must be entered at spreadsheet Layer 2 (with the X.25 personality package installed) in order for a Layer 3 program to have access to the data line. Once this Layer 2 program is entered, Layer 3 can send packets out onto the line and receive packets from the line.

```
LAYER: 2
    STATE: datalink
        CONDITIONS: DL_CONNECT REQ
        ACTIONS: DL_CONNECT CONF
        CONDITIONS: DL_DATA REQ
        ACTIONS: SEND INFO "((DL_DATA)) "
        CONDITIONS: RCV INFO
        ACTIONS: GIVE_DATA
```

   The elements of this program are discussed in Section 34 (OSI Primitives on the Protocol Spreadsheet) and the programming example in Section 37.9.

2. *Valid packet sequencing.* To satisfy RCV conditions, numbered packets must have correct P(R) and P(S) sequencing.

3. *Path.* All RCV conditions that name a packet type may specify one particular "path" as an added condition. Like LCN in DTE and DCE conditions, this path number serves to associate a packet with a particular call. On the X.25 Packet Level Setup screen, up to nine path numbers may be tied to individual sets of

Call Request parameter values, including packet–network "phone" numbers. Refer back to Figure 38-3 for the Packet Level Setup screen.

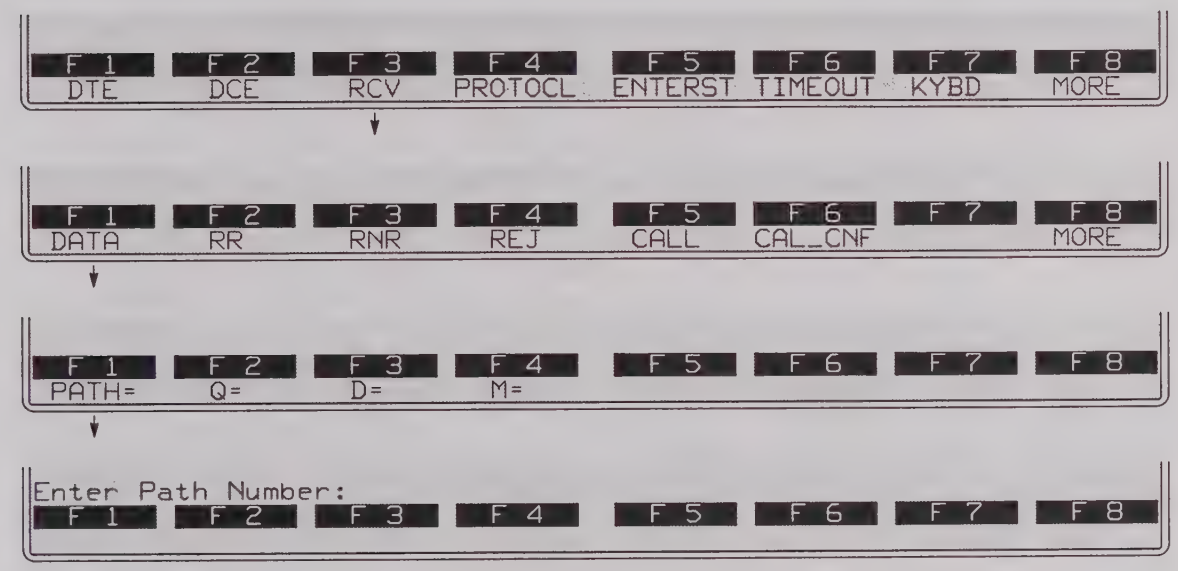


Figure 38-17 *PATH=* replaces *LCN=* as an added condition when the lead is identified as *RCV* rather than *DTE* or *DCE*.

As a packet identifier, the PATH= condition in RCV conditions is more programmable than the LCN= conditions that are available inside DTE and DCE conditions. LCNs usually are assigned dynamically, by the INTERVIEW as well as by other devices, at the time of the call request. By then the test has started running, and it is too late to specify the LCN in the spreadsheet program.

The path number, by contrast, may be preprogrammed on the Protocol Spreadsheet. When the call request is sent or received by the INTERVIEW, the call parameters are correlated to the Packet Level Setup screen. If the INTERVIEW sends a call request that specifies a path number, or if the INTERVIEW receives a call request that matches one of the path entries on the setup screen, the LCN of the call request is tied to the path number (Path 3, say), and any subsequent packets with the same LCN will satisfy packet–type conditions that stipulate PATH= 3.

A RCV condition that specified a path number as a further condition might be the following:

```
CONDITIONS: RCV DATA PATH= 3
```

A data packet would satisfy this condition if (1) it had the same LCN as a call request packet with the Calling, Called, Facilities, and Data fields that are

entered across from Path 3 on the setup screen; and (2) the call still was active. The call–request parameters on the setup screen may refer to calls that originate at the INTERVIEW or to call requests that are incoming.

The PATH= condition is shown in the racks of softkeys in Figure 38-17.

4. *Type invalid or unknown.* RCV conditions can detect packets that are invalid "types"—the packet header is missing, for example, or the LCN is 000 for anything other than a Restart, Registration, or Diagnostic packet. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 38-18.

   A packet may be valid in all respects but have a packet–type field that indicates a nonstandard packet type. Such a packet may be matched by a RCV UNKNOWN condition (Figure 38-18).

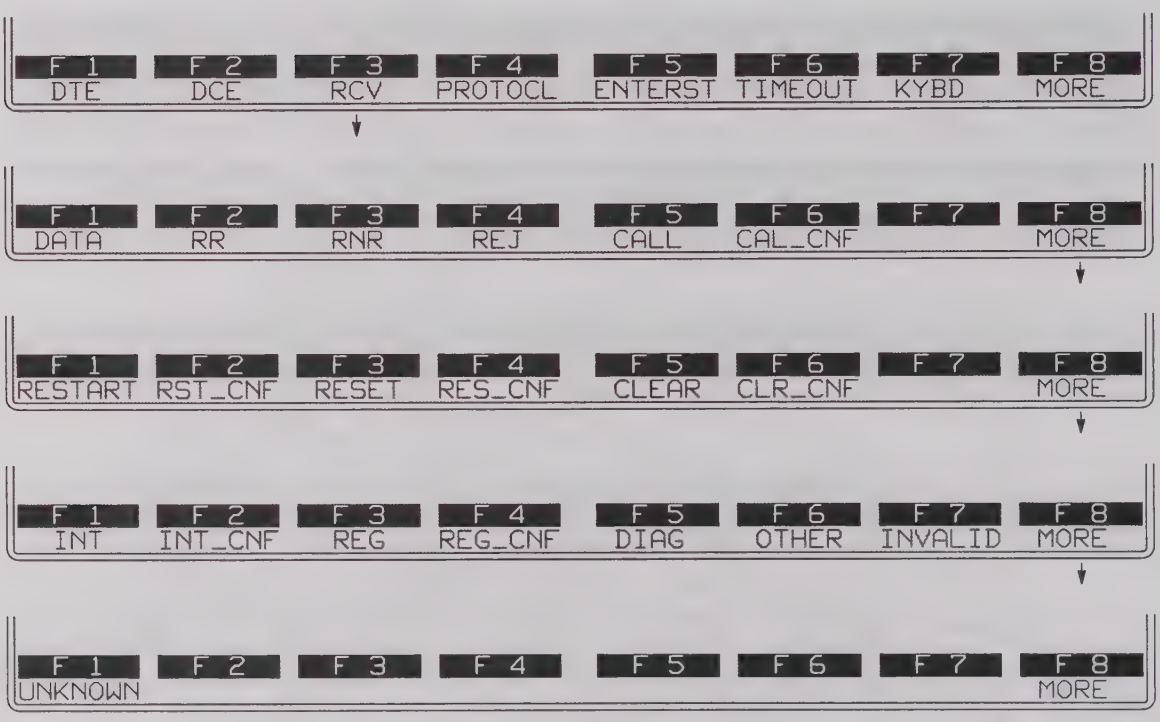


Figure 38-18 *INVALID* and *UNKNOWN* appear on the bottom two racks of *RCV* packet–type softkeys.

(B) P(S) Error

When you emulate at Layer 3, data packets with P(S) errors are detected as PS_ERR conditions, not as RCV DATA conditions.

PS_ERR applies only to packets received when you are emulating. The same packet that triggers a PS_ERR condition also may satisfy a DTE DATA or DCE DATA condition—but not a RCV DATA condition.

PS_ERR will come true for any received data packet whose P(S) value is not one higher than the previous P(S).

In the first rack of condition softkeys at Layer 3, press PROTOCL. Then press the softkey for PS_ERR. See Figure 38-19.

(C) P(R) Error

Received data or supervisory packets may have P(R) errors. Such errors are detected as PR_ERR conditions, not as RCV DATA or RR (or RNR or REJ) conditions.

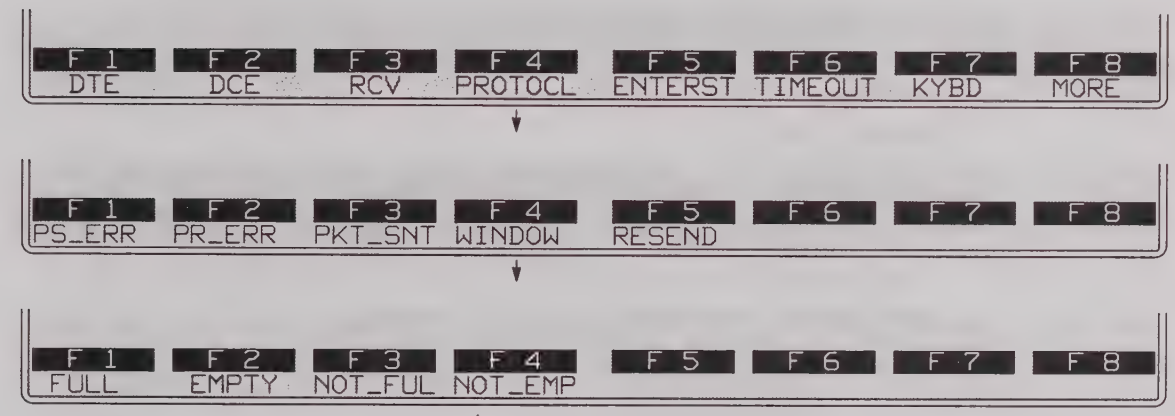


Figure 38-19 The *PROTOCL* key brings up five X.25 Layer 3 emulate conditions.

A valid P(R) is any value that (1) acknowledges a packet that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other P(R) value is detected as an error.

(D) Packet Sent

This condition is true when, as a result of a SEND or RESEND action, a packet has been passed down to Layer 2. This condition may be useful for certain timing measurements, since merely SENDing a packet *does not actually pass the packet down to the next layer* if, for example, the link is not established at Layer 2.

(E) Window Conditions

The size of the Layer 3 window is configured on the X.25 Packet Level Setup screen; see Section 38.1(C). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 38-19.

WINDOW FULL is true when the window is full of unacknowledged packets and the Layer 3 personality package will not buffer additional packets until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no packets outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR
   WINDOW NOT_EMPTY
```

CAUTION: Window conditions are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as a RCV condition.*

(F) More to Resend

Packets in the window may have to be resent, usually as the result of a timeout or a Reject packet. One RESEND action retransmits one packet in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent packets. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
      CONDITIONS: RCV REJ
      NEXT_ST: recover
   STATE: recover
      CONDITIONS: ENTER_STATE
      ACTIONS: RESEND_FIRST
      CONDITIONS: PACKET_SENT
         MORE_TO_RESEND
      ACTIONS: RESEND NEXT
      CONDITIONS: PACKET_SENT
         NO_MORE_TO_RESEND
      NEXT_ST: xfer
```

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 38-20.

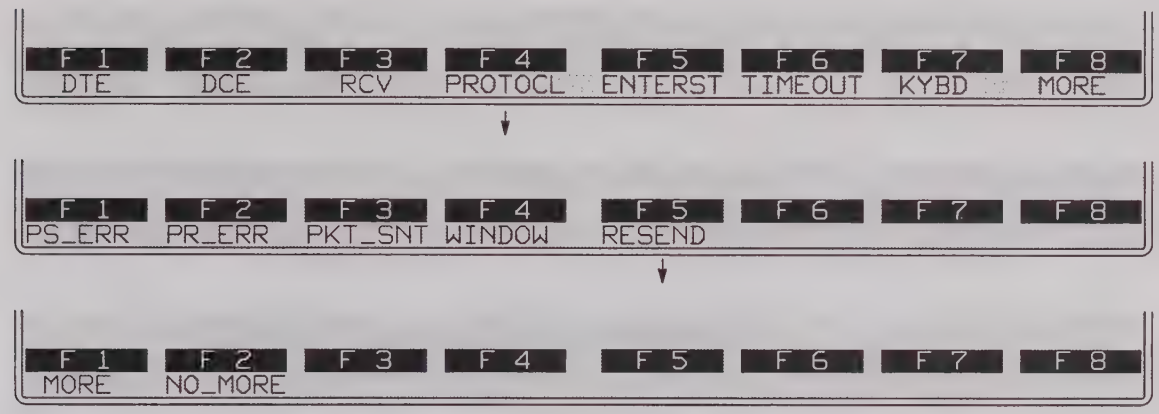


Figure 38-20 The *MORE_TO_RESEND* condition allows you to resend the entire window of packets and then stop when there are *NO_MORE_TO_RESEND*.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as PACKET_SENT.*

38.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 3, press [DONE], then [F4] (ACTION:), to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the X.25 Layer 3 personality package are shown in the four lower racks of softkeys in Figure 38-21. Except for ENHANCE and SUPPRES, the actions shown have meaning only when the INTERVIEW is emulating DTE or DCE, and not when it is monitoring the line passively.

(A) Send Actions

Press the softkey for SEND to access three racks of softkeys with names of packet types that may be named in SEND actions. All data generated by the Layer 3 X.25 package must be enclosed in a packet that is identified in a SEND action by *type*. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of packet types is given in Figure 38-22.

When conditions are true for a SEND action, packets are sent immediately down to Layer 2 to be processed there.

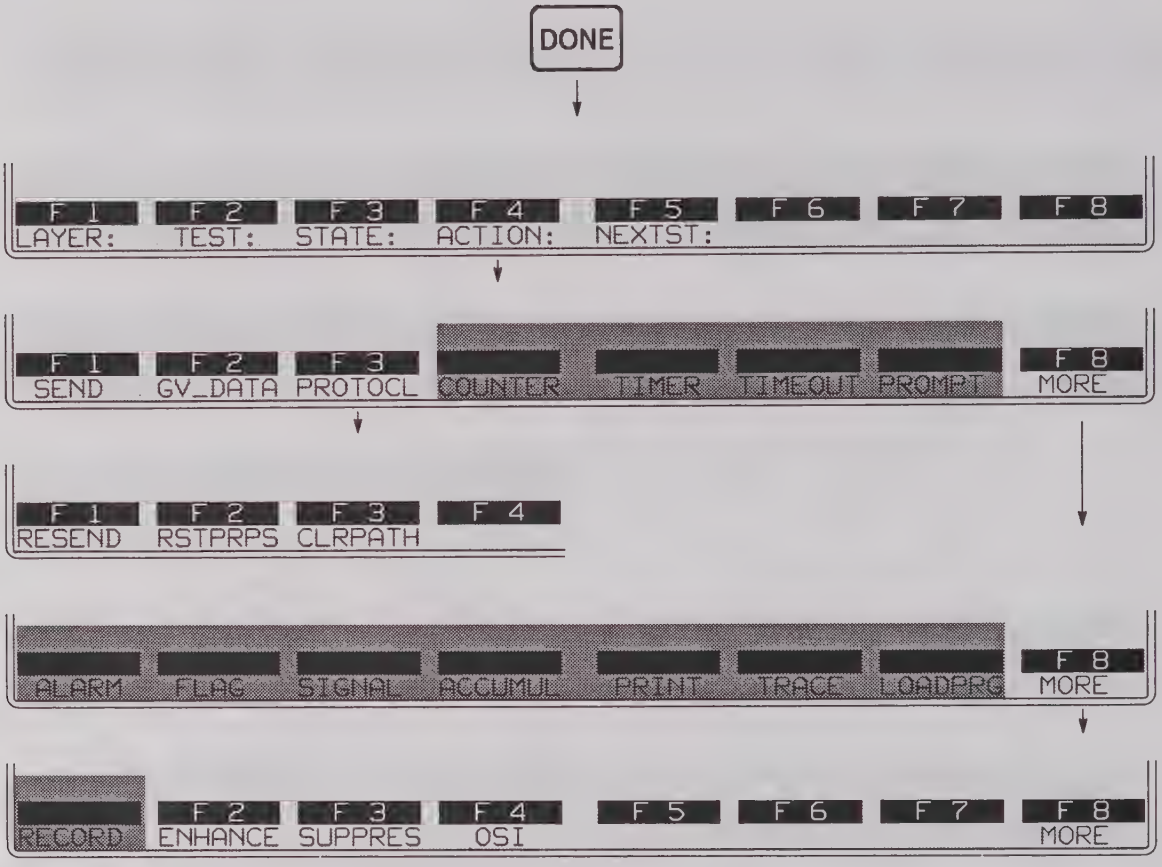


Figure 38-21 Action softkeys specific to X.25 Layer 3.

NOTE: The INTERVIEW is a *layered* emulator. The significance of this is that Layer 3 and higher layers have no direct access (in Emulate modes) to the physical layer, Layer 1. In practice this means that a SEND action at Layer 3 does not in itself send a packet out onto the line. A SEND action merely delivers the packet to Layer 2—provided that Layer 2 has indicated its readiness to receive data from above.

Refer to the Layer 2 program in Section 38.4(A)1. This is the *minimum* program that must be entered at spreadsheet Layer 2 (with the X.25 personality package installed) in order for a Layer 3 program to have access to the data line. Once this Layer 2 program is entered, Layer 3 can send packets directly out onto the line (and receive packets from the line).

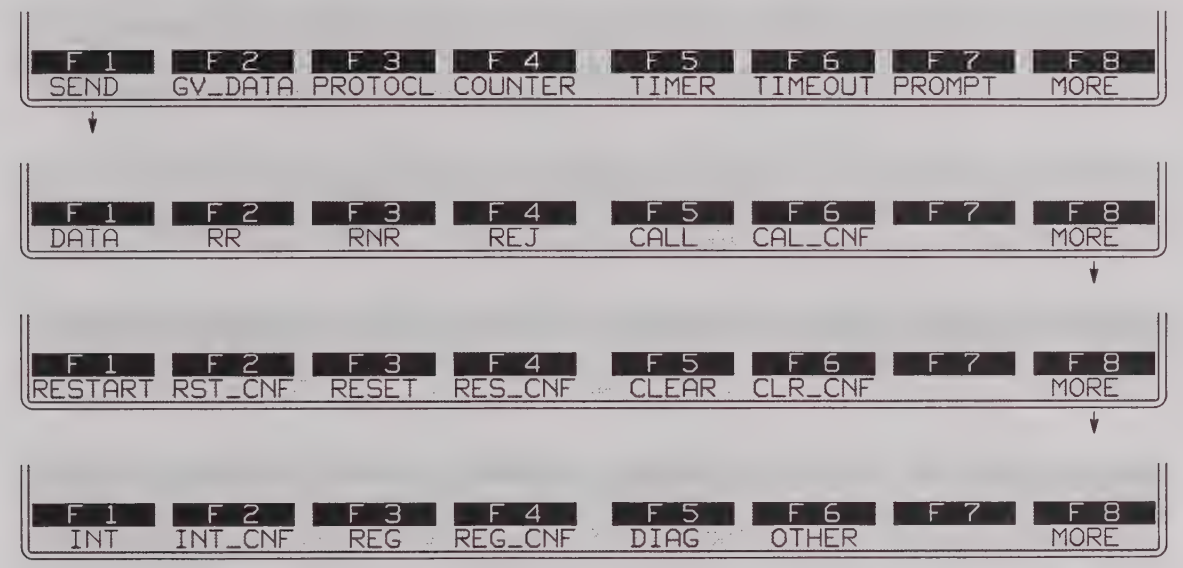


Figure 38-22 *SEND* actions always specify a packet type.

1. *Data packets.* SEND DATA is a complete action–entry. Path, P(S), P(R), Q, D, M, and string parameters may be added to a data packet, but they are optional.

   SEND DATA actions pass the data packet immediately to the next layer down. If the retransmit window is full, the packet is still sent—but it is not buffered in the window and cannot be *re*sent.

   A data packet will be buffered for retransmission regardless of the status of the window if a specific value is entered for the PS= parameter. See *"P(S),"* below. The specific P(S) value will clear the window so that the data packet will be buffered in the first window position.

2. *Supervisory packets.* SEND RR, SEND RNR, and SEND REJ are complete action–entries. Path and P(R) parameters may be added to a supervisory packet, but they are optional.

   Figure 38-23 shows the parameter options for supervisory SEND packets.

3. *Call Request packets.* SEND CALL and SEND CALL_CONF are complete action–entries. Normally a Call Request packet will be entered with additional parameters. Parameters that are available are PATH=, D, FAC=, and STRING.

   When a SEND CALL action does not specify a path, it yields a packet with the LCN that is next in the assignment series: see Section 38.1(D). This is true also when a Call Request specifies a path but the **LCN** column is blank for that path on the X.25 Packet Level Setup screen.

When a PATH= value is included in a SEND CALL or SEND CALL_CONF packet, the packet will be sent with the LCN, Called Address, Calling Address, Facilities, and Data entries that the operator has provided for that path number on the Packet Level Setup screen. A SEND CALL action that is not linked by a PATH= number to a set of call parameters on the Packet Level Setup screen cannot yield a valid call request no matter what string is added to it, since the address–length field (Figure 38-6) in such a packet will be fixed automatically at %.

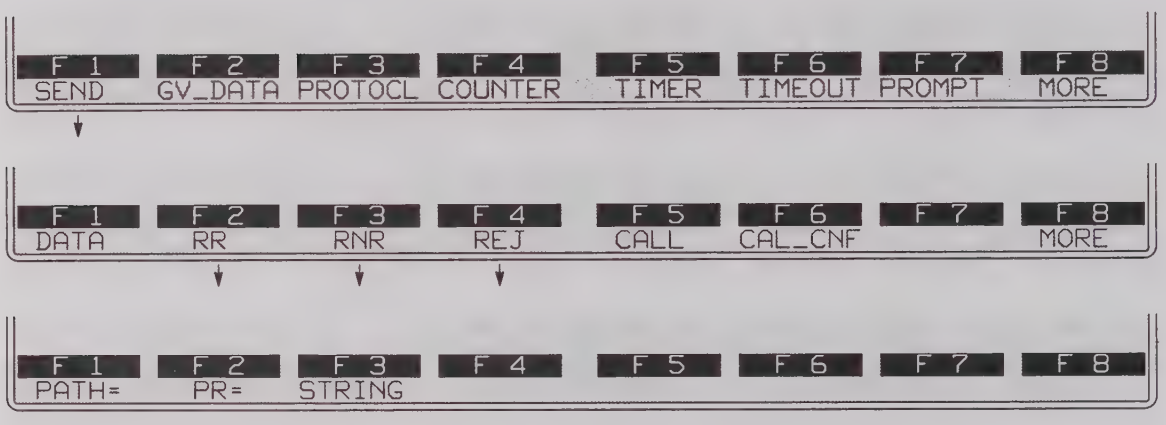


Figure 38-23 Path and P(R) components may be selected for *SEND RR*, *SEND RNR*, and *SEND REJ* actions.

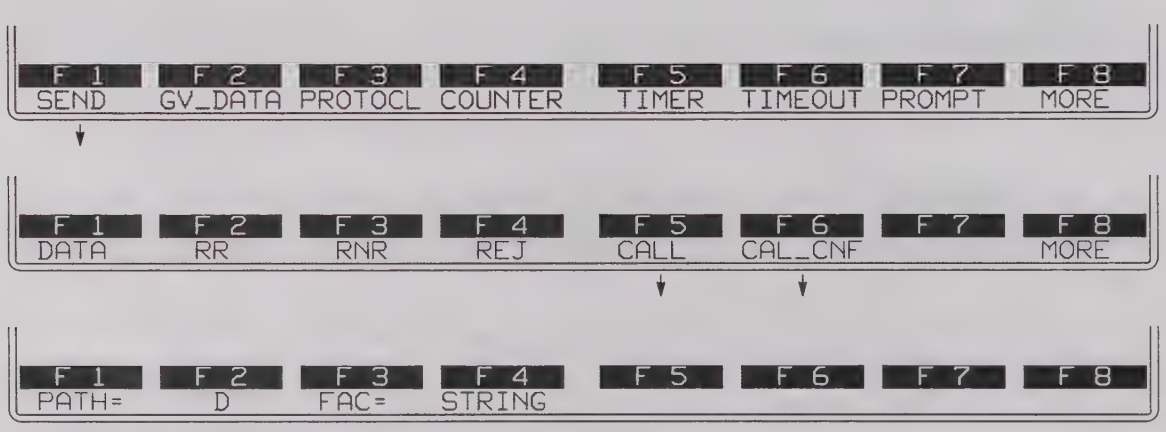


Figure 38-24 *SEND CALL* actions *should* include a path number and *may* set the D–bit; they also may append facilities and a data string to the parameters already listed on the Packet Level Setup menu.

The FAC= option provided on the Protocol Spreadsheet is intended to supplement the **FACILITIES** field on the Packet Level Setup screen. A FAC= string

in a spreadsheet action will be *appended* to the Facilities string on the setup screen. This is in case the facilities entry must be longer than the ten bytes permitted on the setup screen. Do not *repeat* your setup–screen Facilities entry on the Protocol Spreadsheet.

Similarly, any STRING entry on the spreadsheet will be appended to the string in the **DATA** field on the Packet Level Setup screen.

4. *Other unnumbered packets.* The rest of the unnumbered–packet types have softkey options appropriate to their protocol fields (see Figure 38-6). Available softkey parameters for these packet types are PATH=, CAUSE=, DIAG=, and STRING.

Figure 38-25 shows the softkey rack under Registration Confirm, an unnumbered–packet type with the four possible softkey parameters.

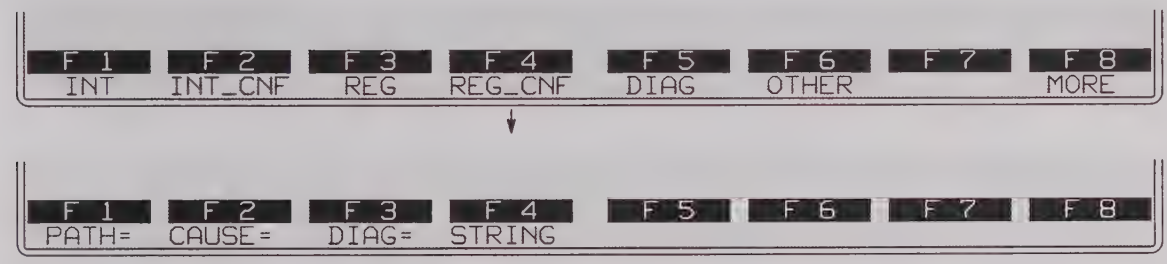


Figure 38-25 Four optional parameters may be added to a *SEND REG_CONF* action.

5. *"Other" packets*. Any packet type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 38-22. Enter the hex value in the form of two alphanumerics. Here is a Call Confirm packet entered as a SEND OTHER action:

```
ACTIONS: SEND OTHER 0F
```

This SEND OTHER 0F action is a good way to send a "stripped down" Call Accepted packet that does not include the additional address and facilities parameters that you may have entered on the Packet Level Setup screen. These parameters sometimes are not used in Call Accepted packets in specific network implementations of X.25.

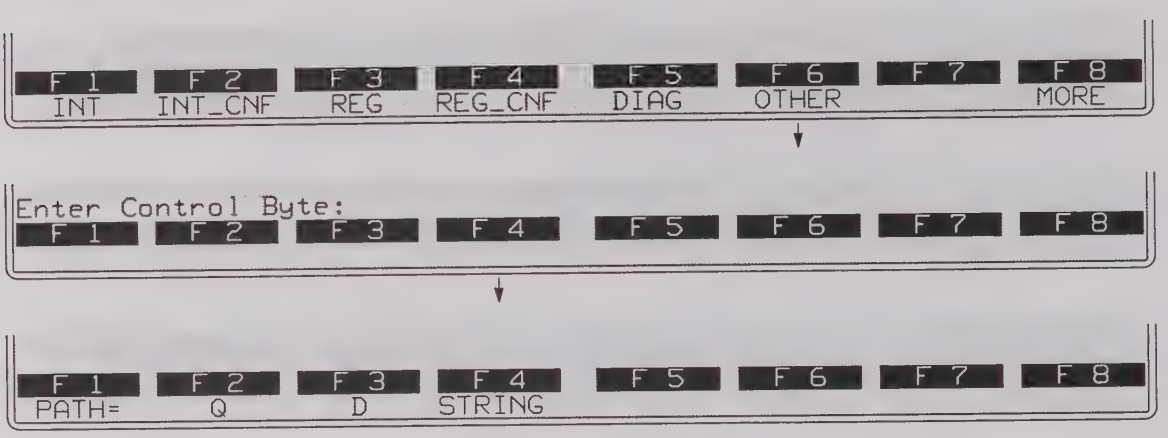


Figure 38-26 Use the *SEND OTHER* action to customize a packet type.

Additional parameters for SEND OTHER actions are PATH=, Q, D, and STRING. (See Figure 38-26.) Since M, P(R), and P(S) fields are embraced already in the user–entered hexadecimal control field, these fields are not given as softkey parameters.

6. *Path.* The addition of a PATH= entry in a SEND action will insure that the packet receives the same LCN as the Call Request with the same PATH= value. The LCN itself is not used for identification in SEND actions, since LCNs usually are assigned dynamically at the time of the call—too late to be programmed on the Protocol Spreadsheet.

   Each path number is tied to a particular set of Call Request parameter values on the X.25 Packet Level Setup screen. See Figure 38-3.

   All packet types permit SEND actions that have PATH= options except Restart, Diagnostic, and Registration. These packets do not refer to a specific call or path. They always receive LCN 000.

   As a general rule, path numbers are used at a given layer in the INTERVIEW if (1) the protocol at that layer is multiaddress or multichannel; or (2) the protocol at a layer *below* the given layer is multiaddress or multichannel. Use the same path number at each layer for a given call.

7. *P(S).* P(S) fields are transmitted in data packets only. (See the packet–field diagrams in Figure 38-6.) The softkeys that open below PS= are illustrated in Figure 38-27.

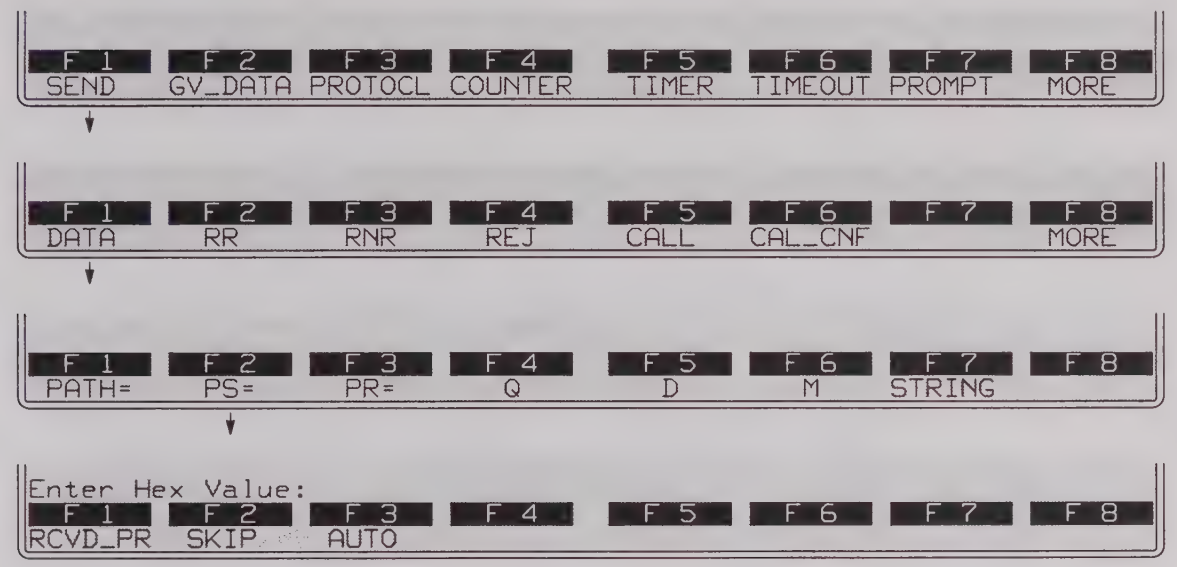


Figure 38-27 The P(S) field may be specified in a *SEND DATA* action.

To specify a P(S) value, press the softkey for PS=, then enter a hexadecimal in the form of one or two alphanumerics. An entry that represented the highest valid P(S) in MOD 8 would be PS= 7. The highest valid entry in MOD 128 is PS= 7F. A SEND DATA action that *specifies* a P(S) value—PS= 0, for example—will clear the window so that the data packet is passed immediately to Layer 2.

Other P(S) options are RCVD_PR, SKIP, and AUTO. RCVD_PR means that you send the P(S) value that the other side says it is expecting. This is the valid P(S) in most cases, but not when you send two or more data packets in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct P(S). This will look to the other side as though a packet has been lost in transmission. This selection causes the window to be cleared.

PS_AUTO is the default setting for SEND DATA actions. AUTO means that every *new* data packet sent will have a P(S) value of one higher than the previous.

8. *P(R)*. To specify a P(R) value, press the softkey for PR= (see Figure 38-28). Enter a hexadecimal value written as one or two alphanumeric digits. Valid hex entries are the same as for P(S).

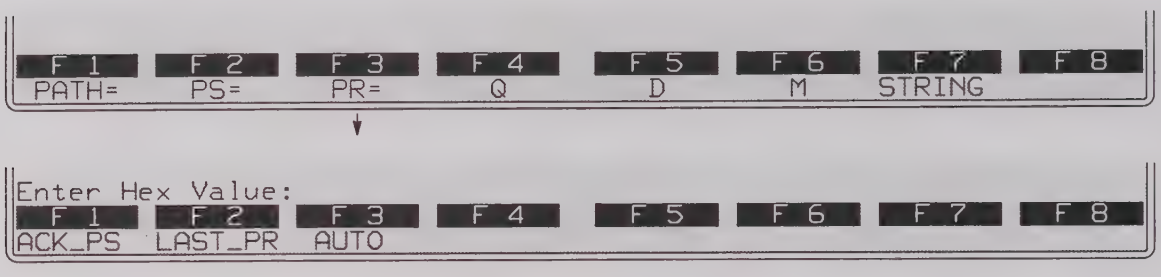


Figure 38-28 The P(R) field may be specified in data and supervisory packets to be sent.

Other P(R) options are ACK_PS, LAST_PR, and AUTO. (See Figure 38-28.) ACK_PS means that your P(R) will acknowledge (that is, it will be *one higher than*) the last P(S) value you received. Normally this will be the correct P(R), except in cases where the last P(S) received was erroneous. The PR= ACK_PS selection allows you to overlook P(S) errors.

LAST_PR means that you simply repeat the last P(R) you sent. Normally this is the correct P(R) following a bad P(S). The PR= LAST_PR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal X.25 Layer 3 PAD, ACKing valid P(S) values and repeating your last P(R) whenever an invalid P(S) is received.

9. *Q, D, and M.* Softkeys for Q, D, and M are included in the full set of optional parameters for a SEND DATA action in the top rack of Figure 38-28.

   Q– and D–bits also may be set in SEND OTHER actions. The D–bit alone is selectable in Call Request and Call Accepted packet types.

10. *Cause.* Actions that send Restart, Reset, Clear, and Registration Confirmation packets may be refined to send a particular cause code and/or diagnostic code.

    Press the softkey for one of these SEND actions, then press [F2], CAUSE. See Figure 38-29. The names of the causes as well as their hexadecimal values are indicated on the softkey–prompt line near the bottom of the screen.

    To specify a particular cause, the user does not have to memorize cause codes or consult a table. Instead, the user presses [F2], ROLL, and repeats the keystroke to cycle through the list of cause names for a given packet type. Figure 38-29 shows the cycle of causes that pertain to Clear Request packets. The user presses [F1], SELECT, when the right cause has "rolled" onto the prompt line. The SELECT softkey writes the current cause onto the spreadsheet screen.

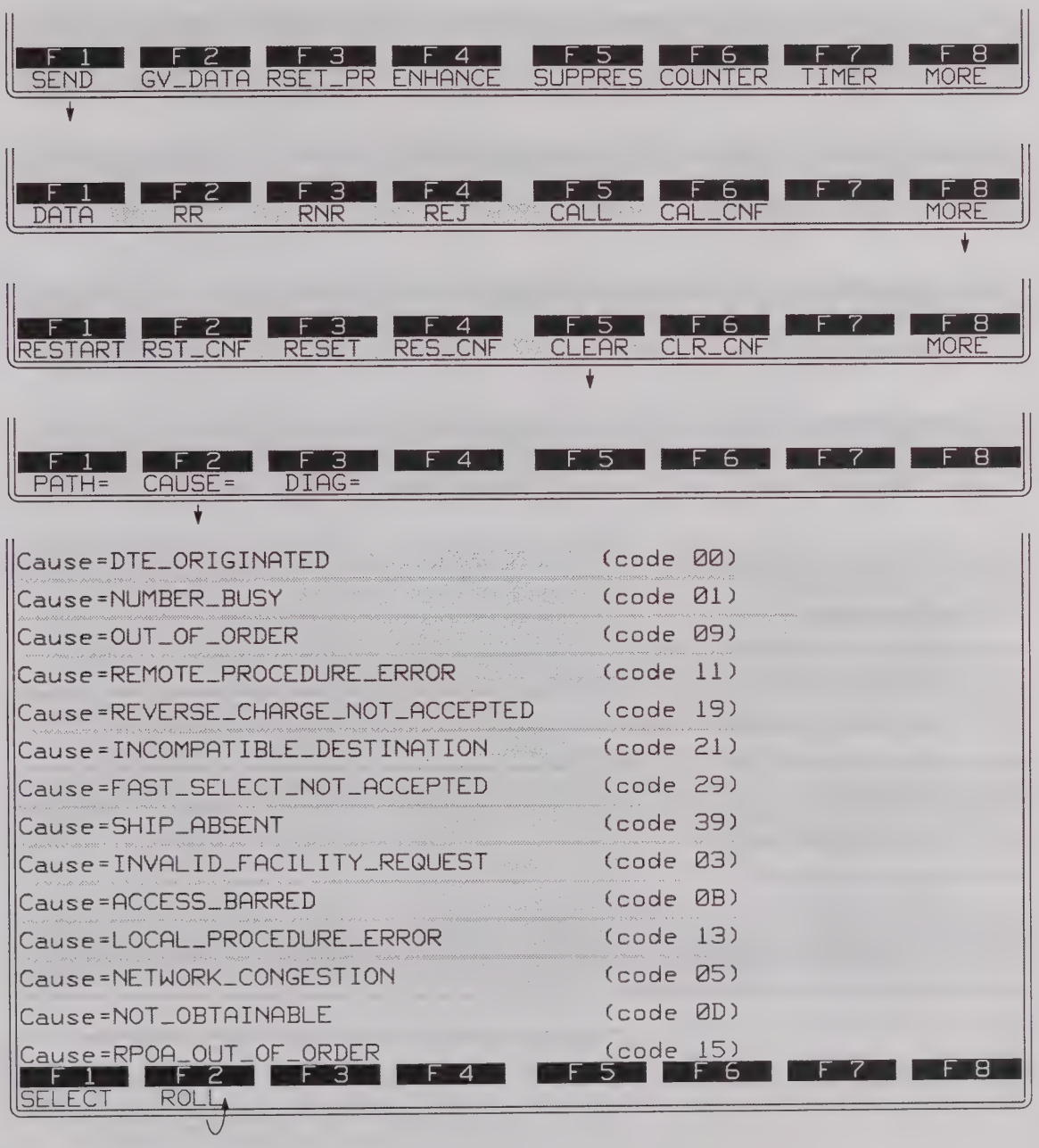


Figure 38-29 These causes are available for a *SEND CLEAR* action.

Causes for Restart, Reset, and Registration Confirmation packets were listed in Figure 38-14 and Figure 38-15.

Here is an example of a cause entry in a SEND action on the Protocol Spreadsheet.

```
ACTIONS: SEND RESTART CAUSE= NETWORK_OPERATIONAL
```

Causes may be entered into the spreadsheet test as numeric values.

11. *Diagnostic.* Two digits representing the one–byte diagnostic–code field (Figure 38-6) may be added to a SEND action for Restart, Reset, Clear, Diagnostic, and Registration Confirmation packets. Refer to F3, DIAG=, in Figure 38-29 or in the lower rack of Figure 38-25.

    Enter the diagnostic code as two hexadecimal digits. Type each digit as an alphanumeric in the range *0–9* and *A–F* (or *a–f*): do not use the HEX key. Here is an example of a SEND action that specifies both a cause and a diagnostic code:

    ```
    ACTIONS: SEND RESET CAUSE= LOCAL_PROCEDURE_ERROR DIAG= 01
    ```

12. *String.* Strings are sent at X.25 Layer 3 only as adjuncts to packet–types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

    Press the SEND softkey followed by the softkey for a packet type. Add any necessary or desired SEND options for the particular packet type. Then press the STRING softkey (see Figure 38-28, for example).

    There is no spreadsheet keyword that identifies send–strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " –character in your string, type \" .)

    Here is a simple SEND action that includes no options besides a string:

    ```
    ACTIONS: SEND INT "0₀"
    ```

    And here is a SEND action that includes a full complement of optional fields, including a string:

    ```
    ACTIONS: SEND DATA PATH= 4 PS= AUTO PR= AUTO "␍␊ This is user
      data."
    ```

    Most ASCII–keyboard, control, and hexadecimal characters are legal in a send–string. Special keys (FLAG, DON'T CARE, BIT MASK) are not legal. Refer to Table 33-2.

    To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)) . Remember that the double parens are special characters produced by the CTRL–9 and CTRL–8 combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

(B) Give Data

GIVE_DATA is the F2 action on the first rack of action softkeys (refer to Figure 38-21). This action takes the data field from a received data packet and passes it up to Layer 4 along with an N_DATA IND primitive. (See Section 34, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data normally arrives up at Layer 4 via GIVE_DATA actions at Layer 3.

(C) Resend

The RESEND function is mapped to F1 on the second layer of action softkeys at Layer 3 for X.25 (below the PROTOCL softkey: see Figure 38-30). A RESEND action will resend the first packet in the window. The window is a queue that buffers data packets for retransmission in case one or more transmissions are lost or in error.

The first packet in the window always is the *earliest outstanding* (unacknowledged) packet. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first–packet" position is established. The first RESEND after an acknowledgment always sends the first window packet.

The *second and subsequent* RESENDs following an acknowledgment also will send the first window packet, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window packets. Figure 38-31 shows the position of the the resend "pointer" after four consecutive RESEND NEXT actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

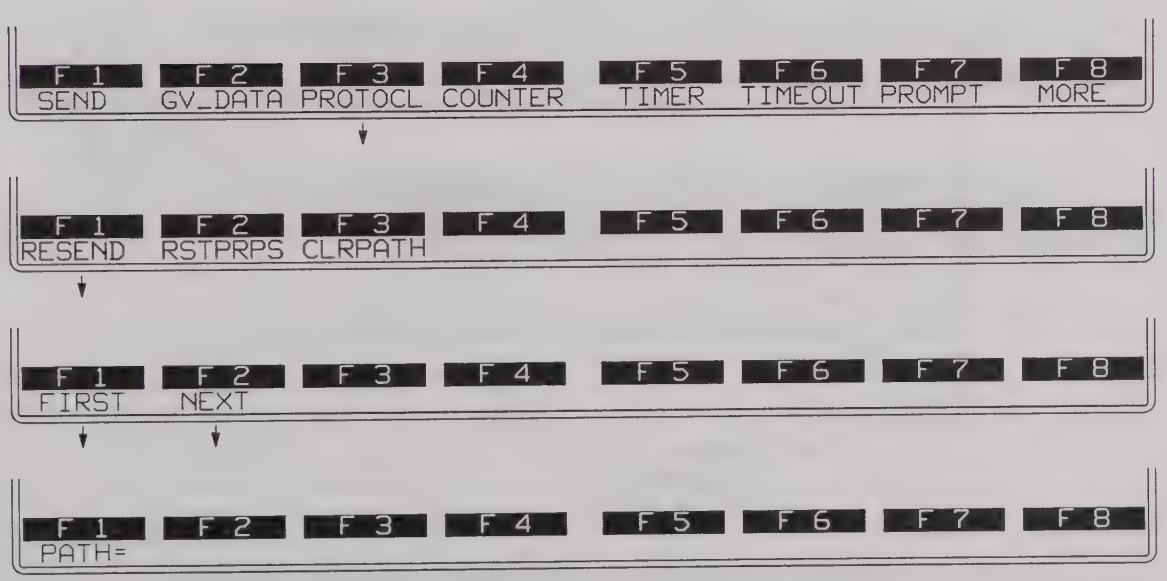


Figure 38-30 The *RESEND* action allows you to recover from sequence errors.

The resend–pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

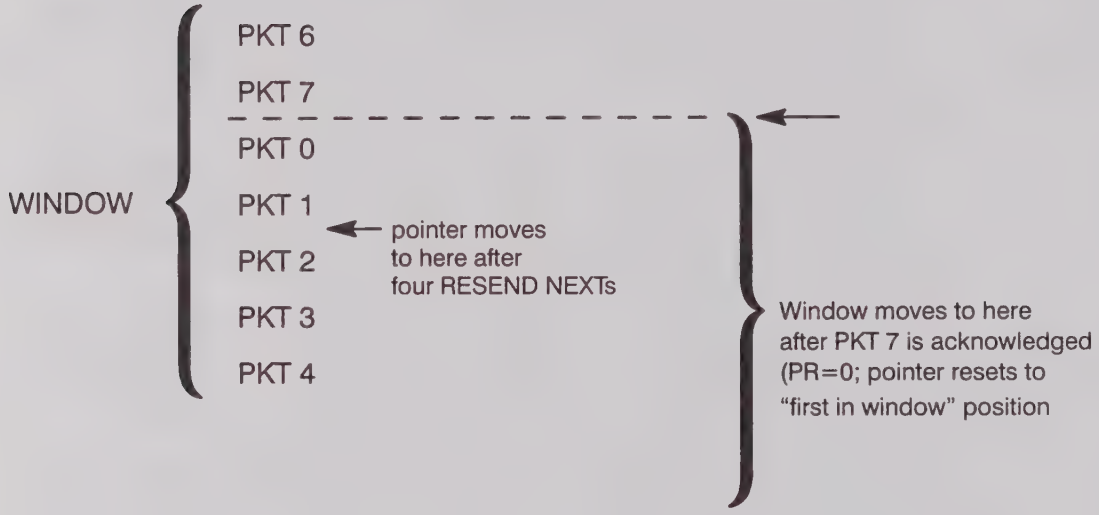


Figure 38-31 Resends cause the pointer to move, while acknowledgments move the pointer *and the entire window*.

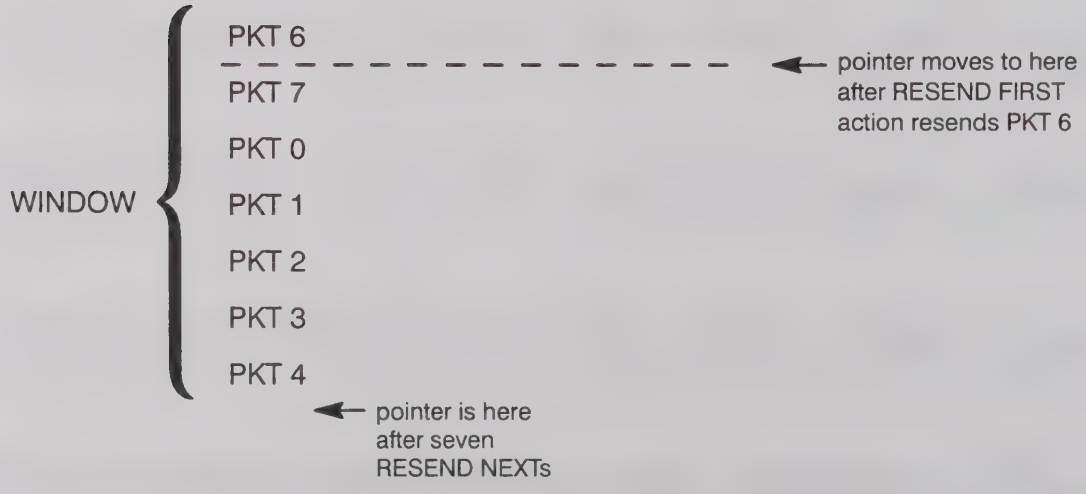


Figure 38-32 *RESEND FIRST* resets the pointer, allowing you to resend the entire window *repeatedly*.

1. *Resend first/next.* RESEND FIRST means that the resend–pointer is reset to the beginning of the window, the first packet in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 38-32.

   The RESEND FIRST action makes it possible for you to resend all the packets in the window one by one, and then *resend them again* if necessary.

2. *Path=* . The path in the resend packet can be set by this optional action. (Default is 0 in a RESEND action.)

(D) Reset P(R) and P(S)

RSTPRPS is the F2 action on the rack of action softkeys below PROTOCL. (Refer again to Figure 38-21.) The sequence–number fields in data packets and supervisory packets can be reset by this Protocol Spreadsheet action. Sequence numbers are not reset *automatically* during a test by any packet that is sent or received.

The path number can be set by an optional PATH= selection.

RSTPRPS also clears the transmit window.

(E) Clear Path

Each call that is established in emulated mode is assigned to one of nine independent "paths," each with its own P(R) and P(S) numbering. Thus, nine LCNs may be active at once. The CLRPATH action (Figure 38-21) allows you to return a path to the pool to be used again.

In the example below, a Clear request is expected. The actions that result will be to send a Clear confirmation and to clear the path.

```
LAYER: 3
   STATE: clearing
      CONDITIONS: RCV CLEAR
      ACTIONS: SEND CLEAR_CONF
         CLRPATH
```

The path number can be set by an optional PATH= selection. Without this selection, the path that is cleared will be that of the most recent packet received.

38.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 3 protocol trace (see Section 38.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 3, the packet that satisfied the condition can be enhanced on the X.25 Layer 3 protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey. Figure 38-33 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

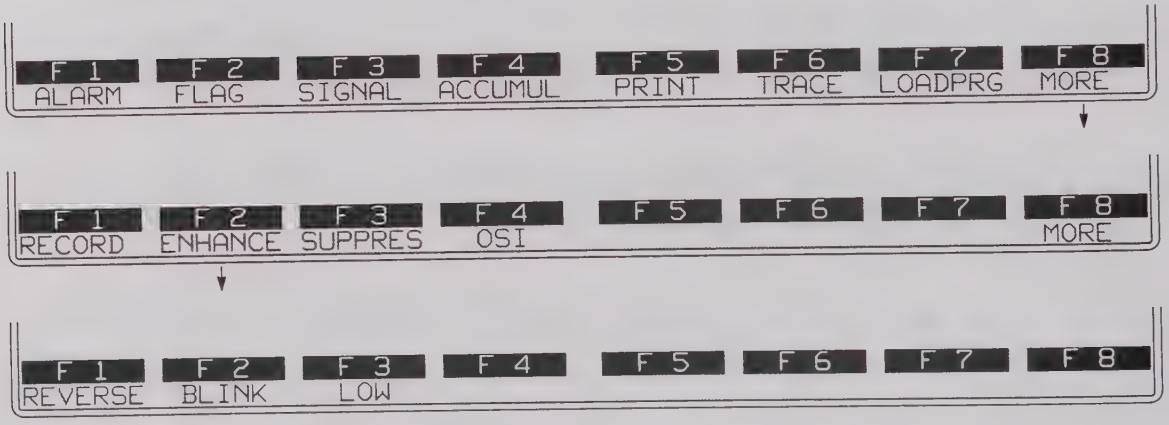


Figure 38-33 Selected packets on the protocol trace may be enhanced or suppressed.

Figure 38-34 shows one screen of a Layer 3 protocol trace in which an Interrupt packet has been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE INT
ACTIONS: ENHANCE REVERSE

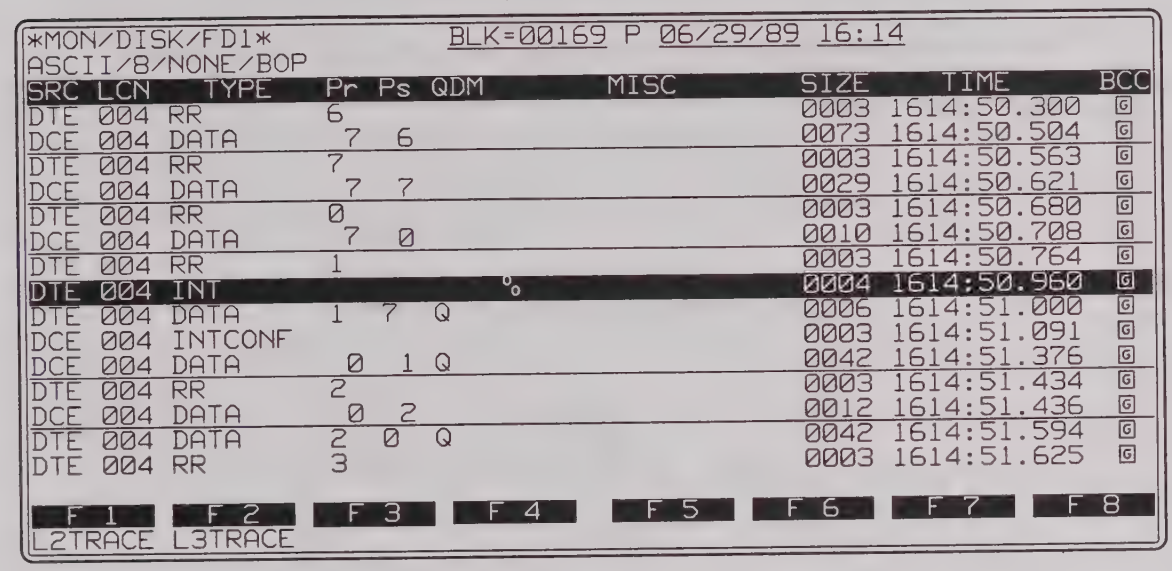


Figure 38-34 An interrupt packet has been highlighted.

(B) Suppress

Individual packets that are suppressed in Layer 3 actions are deleted from the trace display. Figure 38-33 shows the softkey path to SUPPRES.

38.7 Automatic Primitives

In Section 34, Table 34-2 lists the OSI service primitives that are monitored at the boundaries of Layer 3 as trigger conditions and sent up to Layer 4, or down to Layer 2, as user–entered spreadsheet actions. These primitives are layer–specific rather than protocol–specific and are not part of the personality package for X.25 Layer 3; but a few of the primitives are set in motion automatically by X.25 Layer 3 spreadsheet actions. These *automatic* primitives can be thought of as part of the Layer 3 actions themselves, and by extension, as part of the X.25 protocol package.

Table 38-1 gives the set of X.25 Layer 3 actions that have action–primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 3, an N_DATA IND primitive is forwarded to Layer 4, where an N_DATA IND condition may be waiting to monitor it.

Table 38-1
Automatic Primitives Generated at X.25 Layer 3

X.25 Layer 3 Action	Automatic Primitive	To Layer
GIVE_DATA	N_DATA IND	4
SEND {TYPE}	(DL_CONNECT REQ*)	2
	DL_DATA REQ	2
RESEND	(DL_CONNECT REQ*)	2
	DL_DATA REQ	2

*Sent if Layer 2 shows inactive status. DL_DATA REQ delayed until DL_CONNECT CONF returned by Layer 2.

Whenever a SEND or RESEND action is initiated at Layer 3, a DL_DATA REQ primitive is sent downward along with the DL data (the entire packet). This automatic primitive, which was nowhere entered by the user as an action at Layer 3, still will cause a DL_DATA REQ condition to be true at Layer 2.

If a SEND or RESEND action is triggered at Layer 3 while the data link at Layer 2 is inactive, Layer 3 will sense the absence of a link and delay the DL_DATA REQ. Instead, it will send a DL_CONNECT REQ primitive. Only when a DL_CONNECT CONF has been returned by Layer 2 will Layer 3 release the data and the data primitive.

38.8 Programming Example: Forcing Data Packets Out on the Line

This program is constructed around the "line−access" program that was given at the beginning of Section 38.4(A). It has elements in common with the Layer 2 emulation in Section 37.9.

The program allows you to send data packets containing fox messages out onto the line interface (and up on the display) *even when you are not connected to another device*. In other words, it allows you to get the feel of layered programming before you attempt a live emulation.

The bulk of the program is entered at Layer 2. Personality packages for X.25 must be loaded in at Layers 2 and 3.

Sample Test: Force Data−Packet Transmit:

```
LAYER: 3

    STATE: fox

        CONDITIONS: KEYBOARD '' Ff''
        ACTIONS: SEND DATA '' ((FOX)) ''


LAYER: 2

    STATE: LINK

        CONDITIONS: DL_CONNECT REQ
        ACTIONS: DL_CONNECT CONF
        NEXT_STATE: info_xfr

    STATE: info_xfr

        CONDITIONS: DL_DATA REQ
        ACTIONS: SEND INFO '' ((DL_DATA)) ''

        CONDITIONS: RCV INFO
        ACTIONS: SEND RR RESP
            GIVE_DATA

        CONDITIONS: T1_EXPIRED
        NEXT_STATE: xmt_wndw

    STATE: xmt_wndw

        CONDITIONS: ENTER_STATE
        ACTIONS: RESEND FIRST

        CONDITIONS: FRAME_SENT
            MORE_TO_RESEND
        ACTIONS: RESEND NEXT

        CONDITIONS: FRAME_SENT
            NO_MORE_TO_RESEND
        ACTIONS: ALARM
        NEXT_STATE: info_xfr
```

At Layer 3, you simply enter a KEYBOARD condition and a SEND action. During Run mode, you will press the F key in order to send a fox message inside a data packet.

The DL_CONNECT REQ primitive is sent automatically by Layer 3 before it hands the first data packet down to Layer 2. The DL_CONNECT CONF action–primitive is entered "manually" at Layer 2. It is meant to fool Layer 3 into thinking that there is a link.

When Layer 2 does not receive an acknowledgment to its first INFO frame before a T1 timeout expires, it resends the INFO frames containing the data packet (containing the fox message). The RESEND action restarts the T1 timer automatically. Subsequent timeouts will cause additional resends.

Each time the user presses the F key, a new data packet is added to the retransmit window at Layer 2. With each T1 timeout, the entire window is resent.

39 SDLC

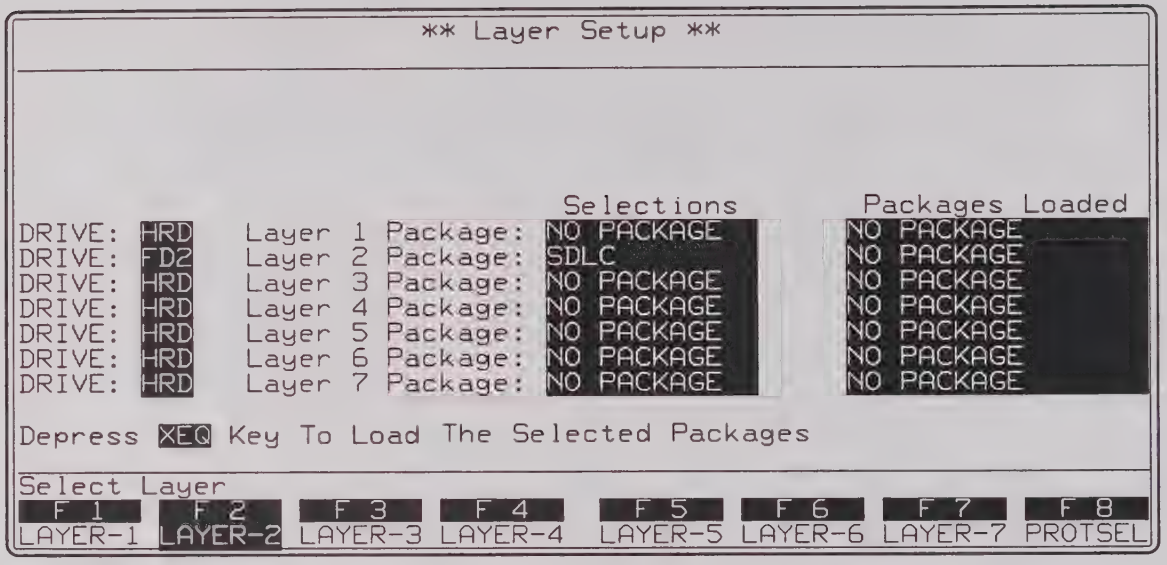


Figure 39-1 The SDLC personality package for Layer 2 is loaded from the Layer Setup screen.

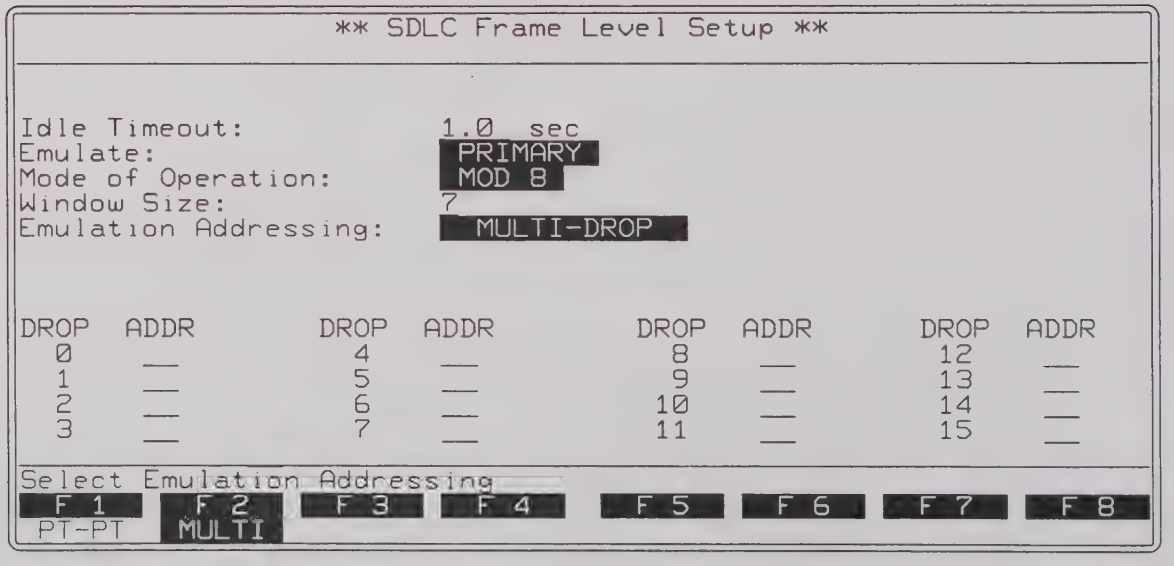


Figure 39-2 Protocol Configuration screen for SDLC.

39 SDLC

SDLC is a "layer personality package" of Layer 2 functions that are loaded into memory from disk via the Layer Setup screen. Figure 39-1 shows the Layer Setup screen configured to load in the SDLC package from floppy−disk, Drive 2. Refer to Section 8 for details on operating the Layer Setup screen.

The SDLC package consists of the following:

- A special SDLC Frame Level Setup screen that controls certain parameters when the unit is tracing or emulating SDLC.
- Multidrop or point−to−point operation.
- A protocol trace (illustrated in Figure 39-3) that displays significant SDLC events. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate SDLC programming. Figure 39-8 shows the softkey path to the first rack of condition softkeys when the SDLC package is loaded in at Layer 2.

39.1 Frame−Level Setup

The parameters on the SDLC Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation. Use this screen also to enable multidrop operation.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the SDLC selection at Layer 2: **SDLC** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up a prompt to **Select Protocol Configuration Screen**. Then press F2 (LAYER−2) to call up the SDLC Frame Level Setup screen.

The parameter fields on this screen are shown in Figure 39-2. **Idle Timeout**, **Emulate**, **Window Size**, **Emulation Addressing**, and **ADDR** apply to interactive (emulate) tests only. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) Idle Timeout

Enter a four–digit (including decimal point) timeout value in this field. The largest valid entry is 65.5 seconds. The smallest entry is .001 second, or 1 millisecond.

Idle timer is the retransmission timer for SDLC INFO and supervisory command frames. When **Emulate:** PRIMARY is selected on the SDLC Frame Level Setup screen and a value is entered in the **Idle Timeout** field on this menu, the Layer 2 package will handle timings as follows:

- Whenever the INTERVIEW sends a command INFO or supervisory frame with the P–bit set and there are no previous polling frames sent by the INTERVIEW currently outstanding (unacknowledged) to the same address, the timer starts timing down from the value entered on the Frame Level Setup screen.
- An acknowledgment by the secondary of the most recent polling INFO or supervisory frame transmitted by the INTERVIEW *stops* the timer (so that it does not expire). This acknowledgment *must* occur in a frame with the F-bit set.
- If F = 0 in the acknowledgment by the secondary of the most recent polling frame sent by the primary, the idle timer restarts at the value selected on the configuration screen.
- An acknowledgment by the secondary of a frame that is not the most recent polling frame transmitted by the INTERVIEW is an incomplete acknowledgment, even if F = 1. This acknowledgment also will restart the idle timer.

Expiration of this Frame Level Setup timeout can only be detected by a T1_EXPIRED condition on the Protocol Spreadsheet at Layer 2.

(B) Emulate Primary/Secondary

There are two selections in the **Emulate** field on the SDLC Frame Level Setup screen, PRIMARY and SECONDARY. The difference between these two modes is that the primary device makes use of the idle timer. The secondary does not.

(C) Mode of Operation

The **Mode of Operation** field refers to the mode of numbering INFO and supervisory frames. There are two options, MOD 8 and MOD 128 .

MOD 8 uses sequence numbers 0–7. MOD 128 adds an extra byte to the control field in INFO, RR, RNR , REJ, and SREJ frames. See Figure 39-5. This extra byte allows sequence numbers in a range of 0–127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate SDLC Layer 2 trace.

(D) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged I−frames that Layer 2 will buffer for retransmission. When the limit is reached, any further INFO frames that are named in SEND actions triggered at Layer 2 will be passed to Layer 1 for transmission but not buffered for retransmission.

The window is a queue that buffers frames for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) frame in the window. Successive RESENDs will send successive frames until there are no more frames to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

(E) Emulation Addressing

Indicate in this field whether the link is POINT−TO−POINT or MULTI−DROP. POINT−TO−POINT is the default selection. For the INTERVIEW to track N(R) and N(S) for multiple addresses, you must choose MULTI−DROP. If you select MULTI−DROP, an additional field, **ADDR**, will appear.

(F) ADDR

Specify the addresses of particular controllers in the **ADDR** table. You may enter up to 16 addresses in the table. The INTERVIEW uses the drop number associated with each listed address to track N(R) and N(S) for resend purposes. All other INFO and supervisory frames with addresses not included in this table will be tracked as a single group. All frames will be displayed on the SDLC protocol trace.

Enter each address as a two−digit hexadecimal value. Use the numbered keys only (not with the [HEX] key) to make these entries. Addresses in the range 00 through FF are valid.

> **NOTE:** If you enter multiple addresses in this table, consider increasing the number of IL buffers. (The default number of buffers is 16.) Use the following formula to determine the number of IL buffers you may need:
>
> number of buffers = ((number of addresses) * (window size)) + 3.
>
> See Sections 28.5 and 62.1 for information on changing the number of IL buffers.

39.2 Protocol Trace

The SDLC package includes an automatic frame–trace display that summarizes link–level activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE ([F2] on the primary rack of display–mode softkeys) to bring the protocol trace for SDLC Layer 2 to the screen.

When running in High–Speed Frame Mode, more data could be passed to Layer 2 than there is room for in the buffer; this will cause an FEB overrun. If this happens, the error message **FE Buffer Overflowed – Some Frames Lost** will appear on the prompt line. The first time an FEB overrun occurs, an audible alarm will also sound; subsequent reccurrences will cause only the message to display (without any alarm). The trace will restart again but some data is lost with each occurrence.

Figure 39-3 is an example of this trace display. Each horizontal row in the trace represents a frame.

(A) The Protocol Trace in Freeze Mode

Press [FREEZE] to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing [ROLL FWD] or [↓] moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing [ROLL BACK] or [↑] moves the viewing window up to add a line of older data to the top of the screen.

Depression of the [NEXT PAGE] key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of [PREV PAGE] adds 15 lines of older frames.

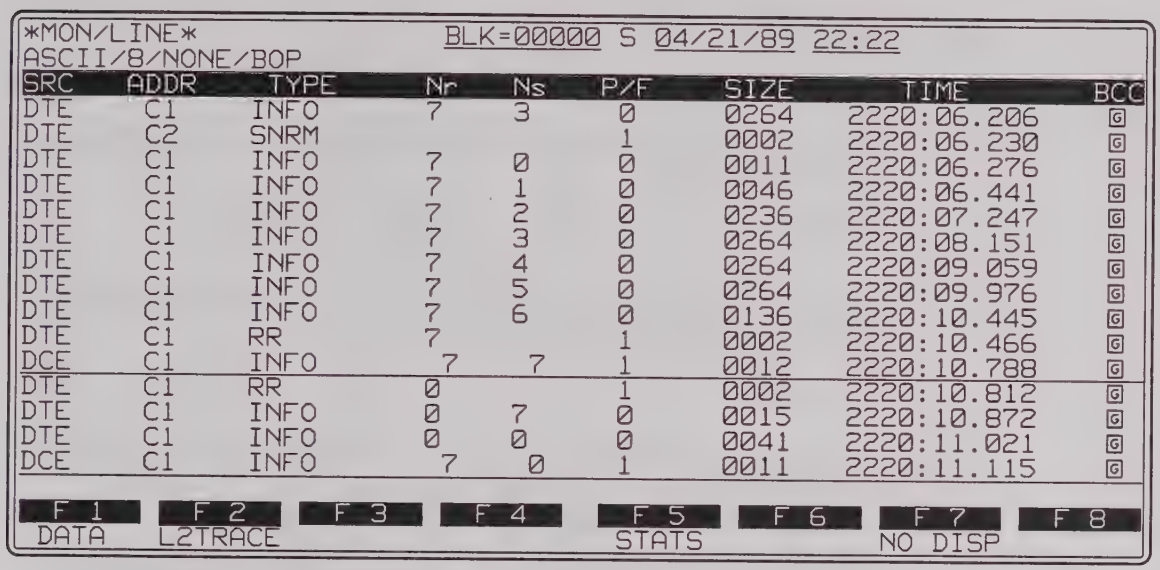


Figure 39-3 Each horizontal row on the trace display represents a frame.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys. Figure 39-4 shows part of a dual–line data screen in Freeze mode. The first frame in the display is the same one that is traced at the top of Figure 39-3.

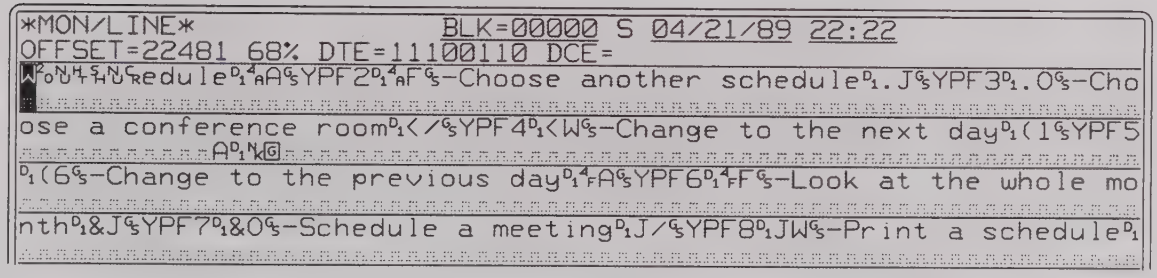


Figure 39-4 Data–display of Protocol Trace shown in Figure 39-3.

(B) Trace Columns

The columns in the protocol trace for SDLC are explained below.

1. *Source*. The **SRC** column identifies the lead on which the frame was monitored, TD (DTE) or RD (DCE).

   Just as on the data display, RD data is underlined.

2. *Address*. The address byte (see Figure 39-5) is given in the **ADDR** column, with its two hexadecimal digits presented as full–size alphanumerics.

   The address in SDLC always belongs to the secondary.

3. *Type*. The mnemonic (abbreviated) names for twenty frame types as they appear in the **TYPE** column of the protocol trace are shown in Figure 39-5 under "CONTROL." The control field, therefore, indicates the frame type. If a control octet does not fit any of the patterns in the figure, the frame is listed in the **TYPE** column as UNKWN followed by the hexadecimal value of the control byte: UNKWN=3F.

   If the number of bytes in the frame is below the required minimum, the frame is posted as INVALID.

4. *N(R) and N(S)*. One column on the frame–level trace is devoted to N(R) values, and one column to N(S). The frame types that include N(R) or N(S) fields in their control fields are indicated in Figure 39-5.

   In multidrop operation, each address listed in the table on the Frame Level Setup screen (Section 39.1) has N(R) and N(S) tracked. All other addresses not included in the table are tracked as a single group.

   N(R) and N(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

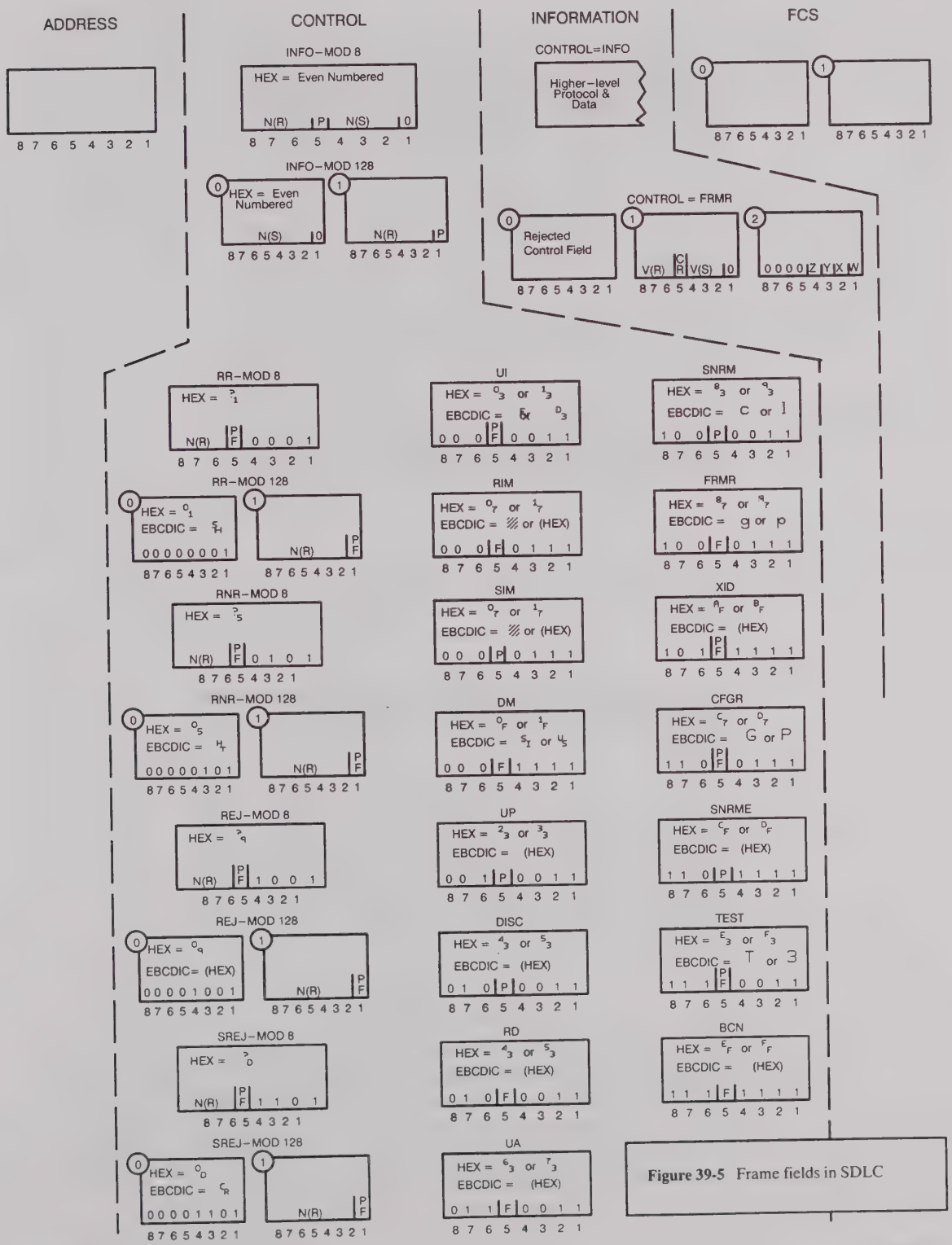


Figure 39-5 Frame fields in SDLC

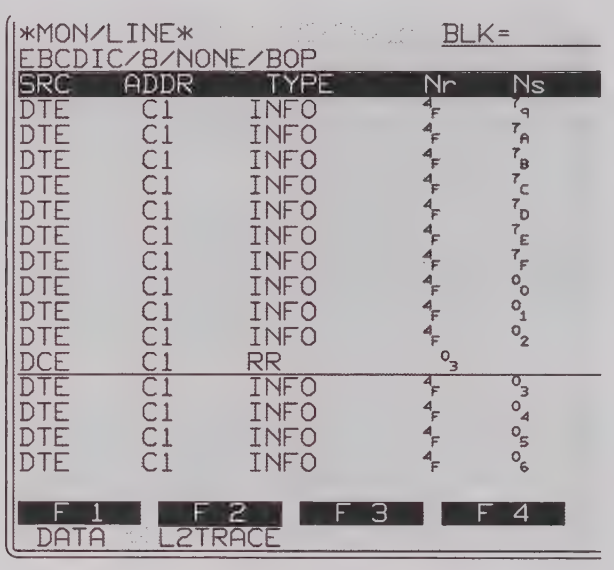


Figure 39-6 MOD 128 sequence numbers are displayed in two–digit hexadecimal *characters*.

N(R) and N(S) values are presented in decimal format in modulo–8 traces. Each column displays a single digit that represents a 3–bit binary value. For modulo 128, the values $^{0}_{0}$ to $^{7}_{F}$ are given in "character" format, where the columns contain a two–digit hexadecimal character (see Figure 39-6).

SRC	ADDR	TYPE	Nr	Ns	P/F
DTE	C1	INFO	7	0	0
DTE	C1	INFO	7	1	0
DTE	C1	INFO	7	2	0
DTE	C1	INFO	7	3	0
DTE	C1	INFO	7	4	0
DTE	C1	INFO	7	5	0
DTE	C1	INFO	7	6	0
DTE	C1	RR	7		1
DCE	C1	INFO	7	7	1
DTE	C1	RR	0		1
DTE	C1	INFO	0	7	0
DTE	C1	INFO	0	0	0

Figure 39-7 Nr and Ns columns are staggered, with the *inside* columns representing the sequence of *DTE* numbered I–frames.

Note that the **Nr** and **Ns** columns on the trace are staggered to suggest four columns. The two inside columns, comprised of the DCE's N(R) value and the DTE's N(S), form a numbering sequence for DTE I–frames. The arrows in Figure 39-7 indicate the sequence: the DTE sends a window full of frames, 0 through 6; the DCE acknowledges frames through 6 (NR=7); the DTE begins a new window with frame 7; and so on.

The two outside columns reveal a similar pattern for DCE I−frames.

5. *P and F.* The status of the poll or the final bit is given in the **P/F** column. Whether this bit is the P− or F−bit is indicated for most frame types in Figure 39-5 (under "CONTROL").

6. *Size.* The number of bytes in each frame is given in this column in four decimal digits. The count begins with the address byte and excludes the two−byte FCS. Frames without I−fields show a count of two.

7. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24−hour clock and posted to the trace display. The clock is accurate to the milli− or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit−image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front−End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front−End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

   If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

8. *Frame checking.* An SDLC frame ends as soon as a ^{7}E flag or seven 1−bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

   A for abort is posted to the displays when a frame is ended by seven 1−bits.

39.3 Monitor Conditions

When the SDLC personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 39-8.

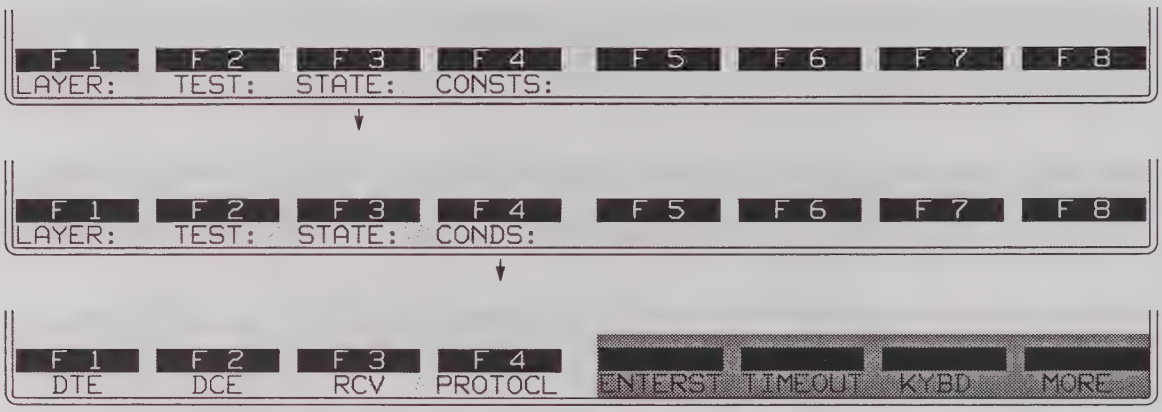


Figure 39-8 Unlike *RCV* conditions, the softkeys for *DTE* and *DCE* are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of frames: INFO, SNRM, UA, and so forth.

(A) Frame Types

The softkeys for INFO, supervisory, unnumbered, and "other" frames are illustrated in Figure 39-9. Press a softkey to write one of these frame types to the Layer 2 spreadsheet. DTE or DCE followed by a frame–type mnemonic—DCE INFO, for example, or DTE SNRM—is a complete condition and will come true if a matching frame is monitored. Address, poll/final, and BCC conditions may be added to the simple frame mnemonic, but they are optional.

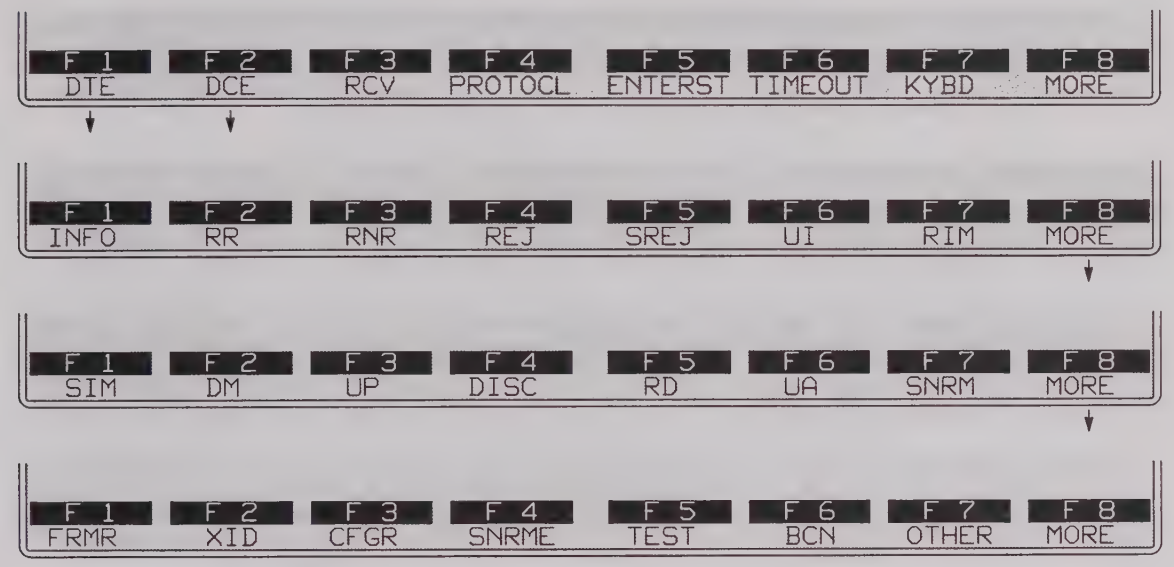


Figure 39-9 Frame types.

1. *Info frames*. INFO frames differ for MOD 8 and MOD 128 numbering schemes. (See Figure 39-5.) For spreadsheet conditions to match I–frames accurately, the correct numbering system ("**Mode of Operation**") should be selected on the Frame Level Setup screen.

2. *Supervisory frames*. The four supervisory–frame types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), REJect, and SREJ (Selective Reject). These frames always contain N(R) fields (see Figure 39-5) and serve mainly to acknowledge or reject INFO frames.

   Like INFO frames, supervisory frames are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

3. *Unnumbered frames*. Unnumbered frames generally assist in link–setup and takedown. These contain neither N(R) nor N(S) fields. Fifteen unnumbered–frame types are shown in Figure 39-9, from UI in the second rack of softkeys through BCN in the bottom rack.

4. *Other frames*. Any frame type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. See Figure 39-10. Then enter the hex byte in the form of two alphanumerics. Here, for example, is a SNRM (with the P–bit set) entered as a hexadecimal:

   CONDITIONS: DCE OTHER 93

Address, poll/final, and BCC conditions may be appended to OTHER conditions. In MOD 8, the P/F bit is already specified in the hex entry, and a P/F condition will be ignored.

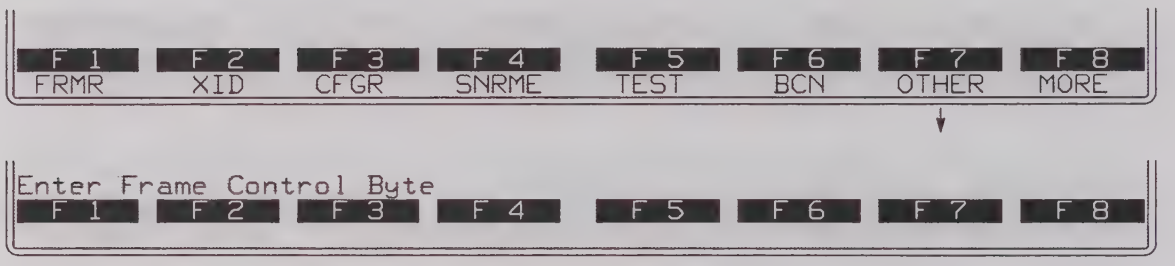


Figure 39-10 The hex value of any frame may be specified under *OTHER*.

5. *Address*. An address condition may be added to INFO, supervisory, unnumbered, and OTHER conditions. Press the softkey for ADR=, shown in Figure 39-11. Then enter the hexadecimal address octet as two alphanumerics. The address octet $^{C}_{1}$, for example, appears as follows:

CONDITIONS: DTE INFO ADR= C1

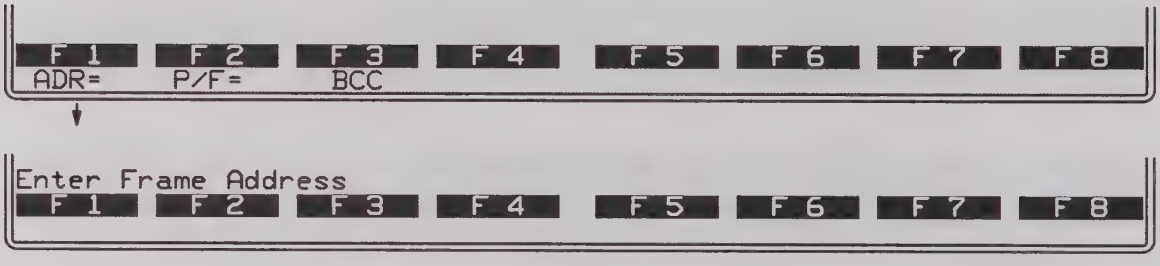


Figure 39-11 The hex value of the address byte is entered as two alphanumerics for *all* frame types.

To bypass the ADR= selection (as well as the other options on the same rack of softkeys in Figure 39-11) press DONE.

6. *Poll/final bit*. P/F conditions are optional for all frame types. P/F values of 0 or 1 are entered by the softkey sequence in Figure 39-12.

   Press DONE to bypass the P/F= condition and the other conditions on the same softkey level in Figure 39-12.

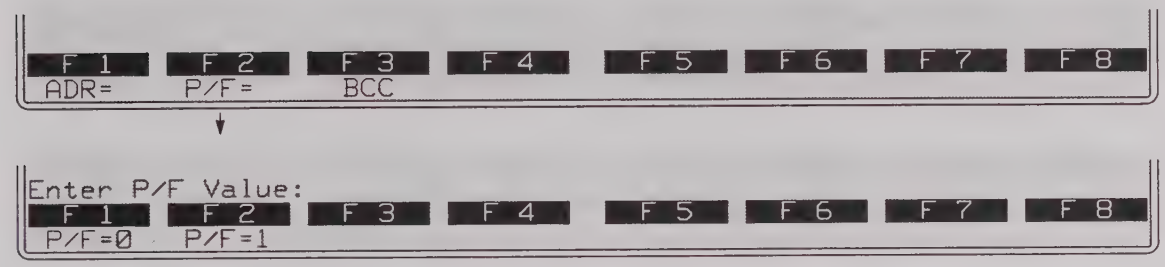


Figure 39-12 The value of the P/F bit may be chosen as a condition.

(B) BCC Conditions

DTE and DCE frames may be monitored for good and bad frame checks and for aborts. *All* DTE or DCE frames may be monitored with respect to frame checking, as in this example:

```
CONDITIONS: DTE BDBCC
```

The softkey sequence for this spreadsheet entry is given in Figure 39-13.

Or a particular *type* of frame may have a BCC or abort condition appended to it:

```
CONDITIONS: DCE INFO ABORT
```

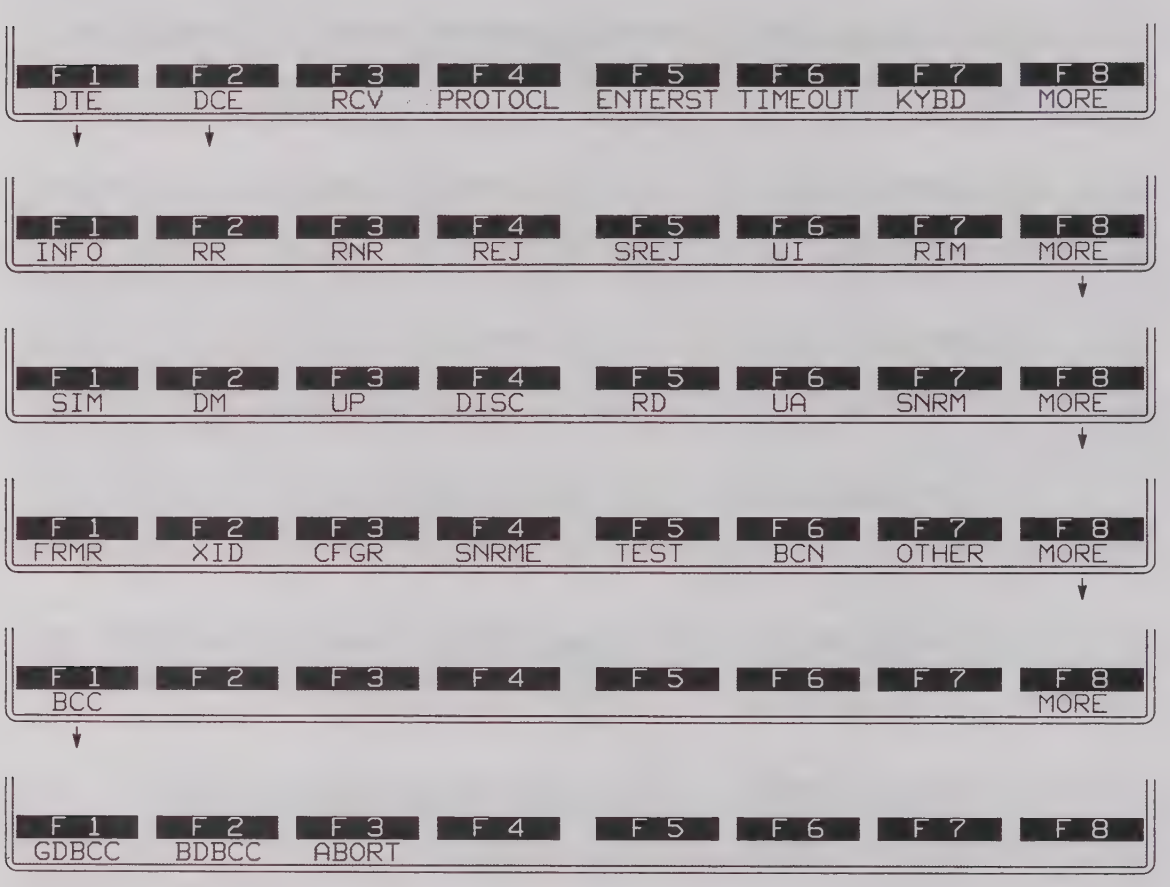


Figure 39-13 A condition may search for *all* good, bad, or aborted frames.

39.4 Emulate–Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for **Mode:** EMULATE DTE or EMULATE DCE.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data lead for SDLC frame types. RCV conditions operate only in emulate modes, and they check only the data lead that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV INFO P/F=1 looks the same as DCE INFO P/F=1—there are important differences that are noted below.

1. *Valid frame sequencing.* To satisfy RCV conditions, numbered frames must have correct N(R) and N(S) sequencing.

2. *Good BCC.* RCV conditions cannot match frames with bad frame checks, nor can they match aborted frames. (Emulate–mode conditions are designed for ease of programming, and the assumption is that as an SDLC emulator, you are not required to acknowledge—or negative–acknowledge—bad or aborted frames.)

   If you wish to count bad BCCs or aborts, use DTE or DCE conditions instead of RCV conditions.

3. *Type invalid or unknown.* RCV conditions can detect frames that are invalid "types"—the control field is missing, for example, or the I–field is missing in an I–frame. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 39-14.

   A frame may be valid in all respects but have a control field that indicates a nonstandard frame type. Such a frame may be matched by a RCV UNKNOWN condition (Figure 39-14).

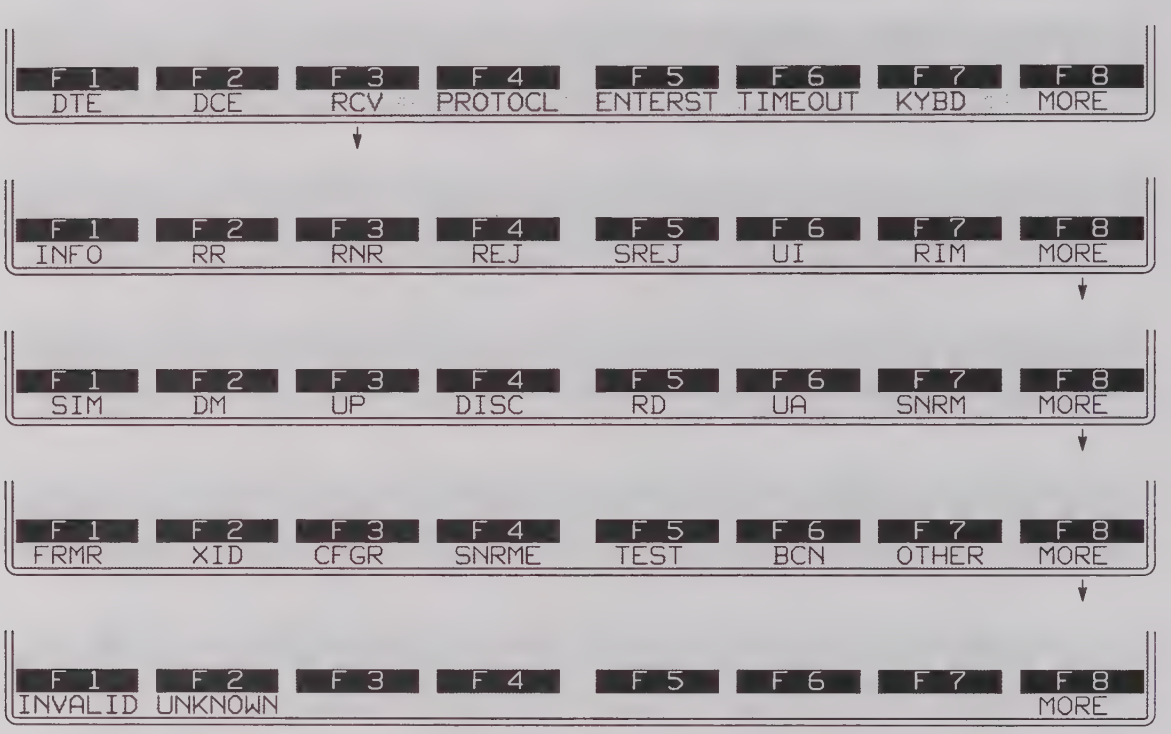


Figure 39-14 *INVALID* and *UNKNOWN* are frame types for *RCV* conditions.

(B) N(S) Error

As a Layer 2 emulator, you do respond to INFO frames that have N(S) errors. These are detected as NS_ERR conditions, not as RCV INFO conditions.

NS_ERRs apply only to frames received when you are emulating. The same frame that triggers an NS_ERR condition also may satisfy a DTE INFO or DCE INFO condition—but not a RCV INFO condition.

NS_ERR will come true for any received frame whose N(S) value is not one higher than the previous N(S).

In the first rack of condition softkeys at Layer 2, press PROTOCL. Then press the softkey for NS_ERR. See Figure 39-15.

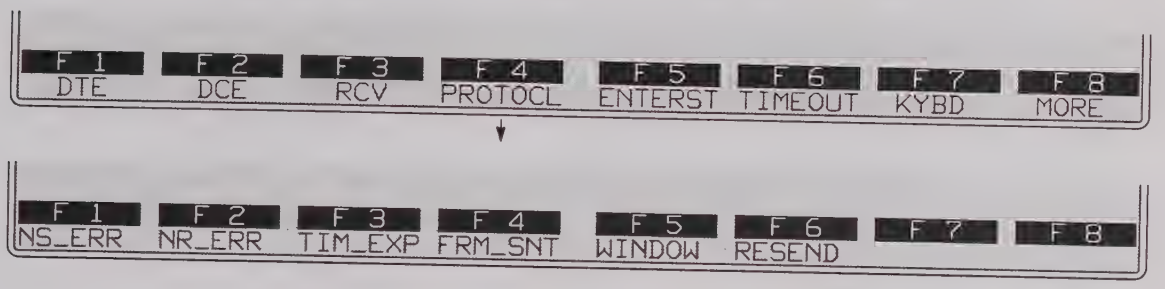


Figure 39-15 The *PROTOCL* key brings up six SDLC emulate conditions.

(C) N(R) Error

Received INFO or supervisory frames may have N(R) errors. Such errors are detected as NR_ERR conditions, not as RCV INFO or RR (or RNR or REJ) conditions.

A valid N(R) is any value that (1) acknowledges a frame that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other N(R) value is detected as an error.

(D) Timeout Expired

This condition detects the expiration of the idle timeout that is regulated on the SDLC Frame Level Setup screen. See Section 39.1(A), above.

(E) Frame Sent

This condition is true when, as a result of a SEND or RESEND action, a frame has been passed down to Layer 1.

(F) Window Conditions

The size of the Layer 2 retransmit window is configured on the SDLC Frame Level Setup screen. See Section 39.1(D). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 39-16.

WINDOW FULL is true when the window is full of unacknowledged frames and the Layer 2 protocol package will not buffer additional frames until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no frames outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR
   WINDOW NOT_EMPTY
```

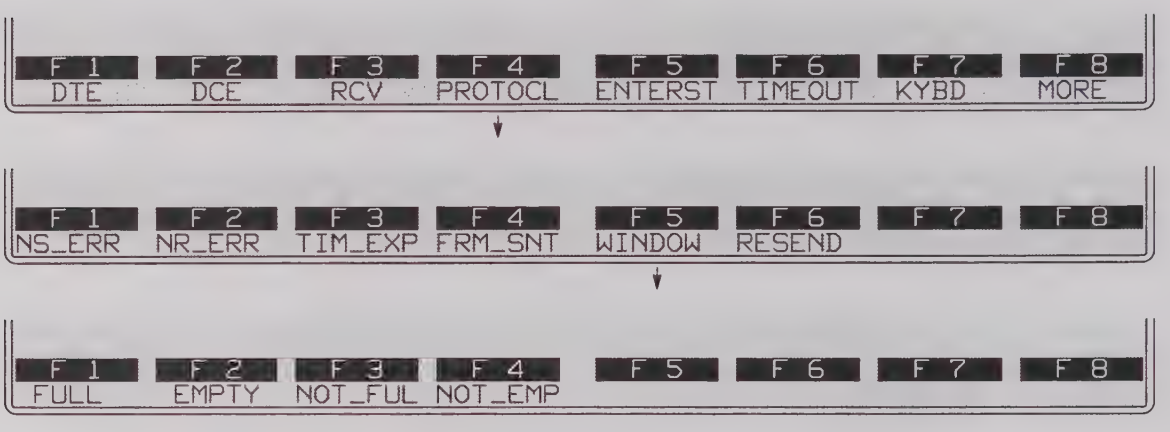


Figure 39-16 When the retransmit window fills, Layer 2 stops passing frames down to Layer 1.

If you select a window condition when the INTERVIEW is configured for multidrop operation, an ADR= softkey selection appears. During multidrop emulation, there may be several transmit windows—one for each controller address listed in the **ADDR** table on the SDLC Frame Level Setup screen. (See Section 39.1.) The INTERVIEW can check window conditions for any of these addresses, but only if you complete the ADR= entry.

> *CAUTION: Window conditions are status conditions (see Section 31.2) and* must *always be used in combination with a transitional condition such as a RCV condition.*

(G) More to Resend

Frames in the window may have to be resent, usually as the result of an idle–timer timeout or a Reject frame. One RESEND action retransmits one frame in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent frames. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
    CONDITIONS: RCV REJ
    NEXT_ST: recover
STATE: recover
    CONDITIONS: ENTER_STATE
    ACTIONS: RESEND
    CONDITIONS: FRAME_SENT
        MORE_TO_RESEND
    ACTIONS: RESEND
    CONDITIONS: FRAME_SENT
        NO_MORE_TO_RESEND
    NEXT_ST: xfer
```

During multidrop emulation, there may be several transmit windows—one for each controller address listed in the **ADDR** table on the SDLC Frame Level Setup screen. (See Section 39.1.) For MORE_TO_RESEND and NO_MORE_TO_RESEND conditions, the INTERVIEW will check the transmit window of the address specified in the last RESEND action.

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 39-17.

> *CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 31.2) and* must *always be used in combination with a transitional condition such as FRAME_SENT.*

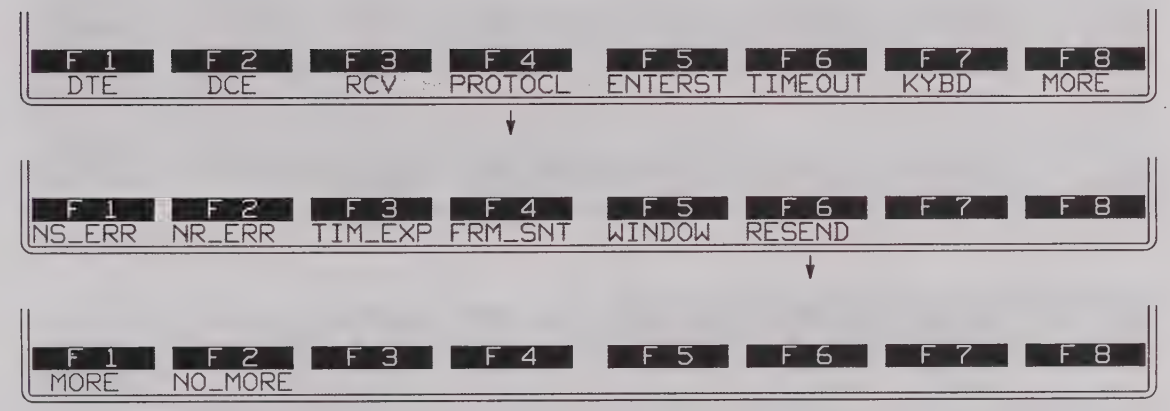


Figure 39-17 The *MORE_TO_RESEND* condition allows you to resend the entire window of frames and then stop when there are *NO_MORE_TO_RESEND.*

39.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 2, press DONE to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the SDLC personality package are shown in the racks of softkeys in Figure 39-18. Except for ENHANCE and SUPPRES, the actions shown have meaning only when the INTERVIEW is emulating DTE or DCE, and not when it is monitoring the line passively.

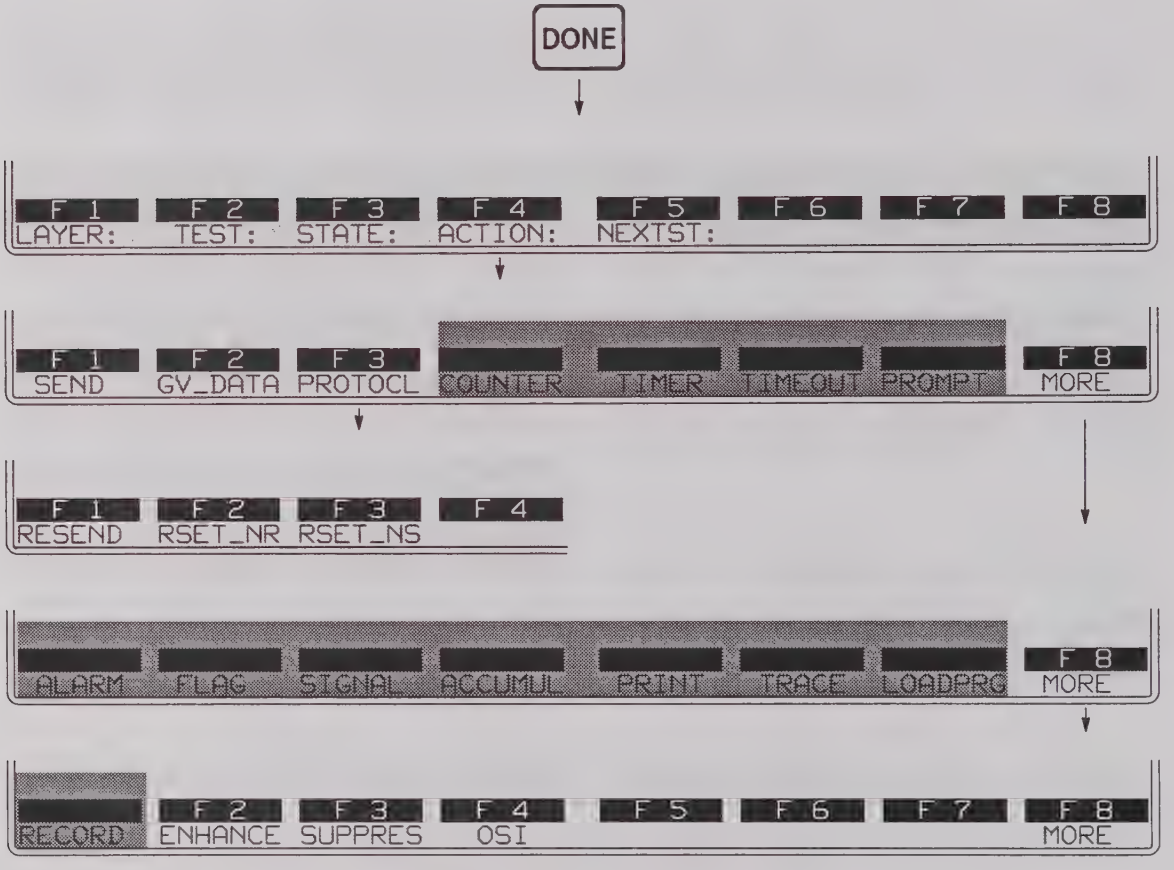


Figure 39-18 Action softkeys specific to SDLC.

(A) Send Actions

Press the softkey for SEND to access three racks of softkeys with names of frame types that may be named in SEND actions. All data generated by the SDLC package must be enclosed in a frame that is identified in a SEND action by *type*. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of frame types is given in Figure 39-19.

When conditions are true for a SEND action, frames are sent immediately down to Layer 1 to be transmitted there.

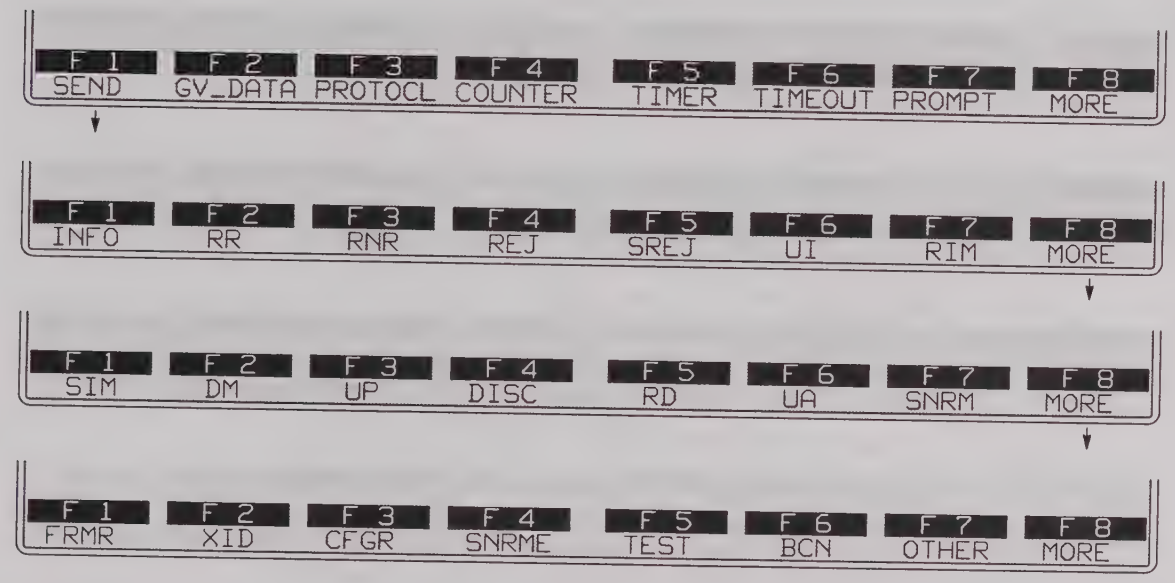


Figure 39-19 *SEND* actions always specify a frame type.

1. *INFO frames*. SEND INFO is a complete action–entry. Poll–bit, N(R), N(S), string, and BCC parameters may be added to an INFO frame, but they are optional. Although not strictly required, you should enter an address.

   SEND INFO actions pass the INFO frame immediately to the next layer down. If the retransmit window is full, the frame is still sent—but it is not buffered in the window and cannot be *re*sent.

   An INFO frame will be buffered for retransmission regardless of the status of the window if a specific value is entered for the NS= parameter (see "*N(S)*," below). The specific N(S) value will clear the window and the INFO frame will be buffered in the first window position.

2. *Unnumbered frames*. SEND SNRM, SEND UA, and so forth, are complete action–entries. P/F–bit, string, and BCC parameters may be added to the SEND action, but they are optional. Although not strictly required, you should enter an address.

3. *Supervisory frames*. SEND RR, SEND RNR, SEND REJ, and SEND SREJ are complete action–entries. P/F–bit, string, and BCC parameters may be added to the SEND action, but they are optional. Although not strictly required, you should enter an address.

The address in SDLC always belongs to the secondary.

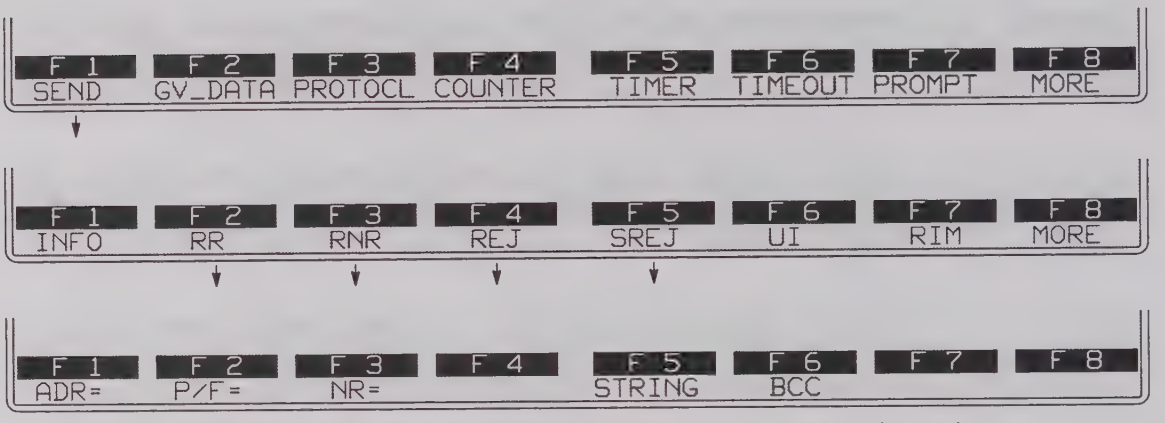


Figure 39-20 Address, P/F, N(R), and block–check parameters as well as data strings may be added to *SEND SUPERVISORY* actions.

4. *Other frames*. Any frame type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 39-19. Enter the hex value in the form of two alphanumerics. Here is a DISConnect command entered as a SEND OTHER action:

   ACTIONS: SEND OTHER 43 ADR= C3

   Since P/F, N(R), and N(S) fields are implied already in the user–entered hexadecimal control field, BCC is a *valid* optional parameters in a SEND OTHER action. (In MOD 128, P/F is not included in the hex entry and is a valid optional entry.) Although not strictly required, you should enter an address.

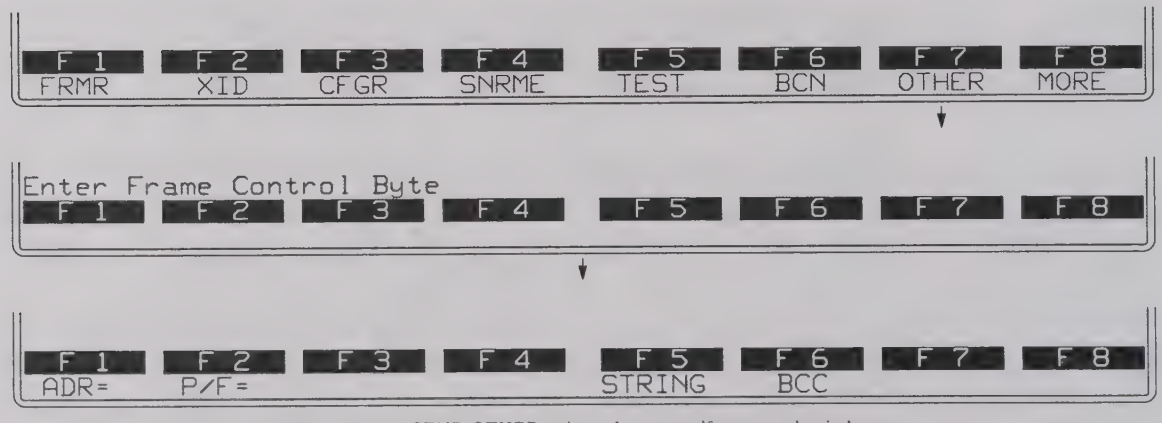


Figure 39-21 *SEND OTHER* actions always specify a *type* value in hex.

5. *Address*. Although not strictly required, an address should be specified for all frame types. The ADR= entry is followed by the hexadecimal address octet typed

as two alphanumeric characters. The address field C3, for example, appears as follows:

ACTIONS: SEND RR ADR= C3

There is also a LOOPBAK softkey selection for the address field. This selection causes the SEND action to refer to the address of the most recently received frame.

If you selected **Emulation Addressing:** MULTI-DROP on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 39.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can send frames to any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will send the appropriate frame from that address' transmit window. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not send to any address.

When **Emulation Addressing:** POINT-TO-POINT is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

6. *Poll/final bit*. The P/F−bit is an optional entry in all SEND actions. A P/F value of LOOPBAK, 0, or 1 are entered by the softkeys in Figure 39-22. If P/F= LOOPBACK, the bit will echo the last P/F−bit received. (Looping the P/F−bit is appropriate for UAs and supervisory frames.)

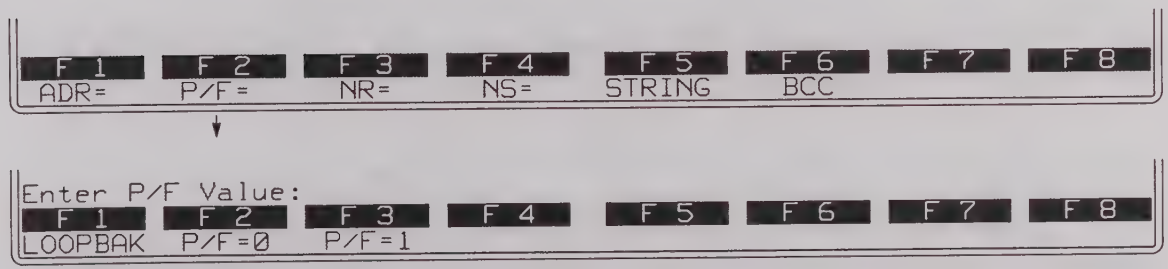


Figure 39-22 A P/F value is optional in all *SEND* entries.

7. *N(R)*. N(R) fields are transmitted in INFO and supervisory frames.

To specify an N(R) value, press the softkey for NR= (see Figure 39-23). Enter a hexadecimal value written as one or two alphanumeric digits. For example, an entry that represented the highest valid N(R) in MOD 8 would be NR= 7. The highest valid entry in MOD 128 would be NR= 7F.

Other N(R) options are ACK_NS, LAST_NR, and AUTO. (See Figure 39-23.) ACK_NS means that your N(R) will acknowledge (that is, it will be *one higher than*) the last

N(S) value you received *for the same frame address*. Normally this will be the correct N(R), except in cases where the last N(S) received was erroneous. The NR= ACK_NS selection allows you to overlook N(S) errors.

LAST_NR means that you simply repeat the last N(R) you sent to the same address. Normally this is the correct N(R) following a bad N(S). The NR= LAST_NR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal SDLC station, ACKing valid N(S) values and repeating your last N(R) whenever an invalid N(S) is received. AUTO is the default N(R) selection in SEND INFO, SEND RR, SEND REJ, and SEND SREJ actions.

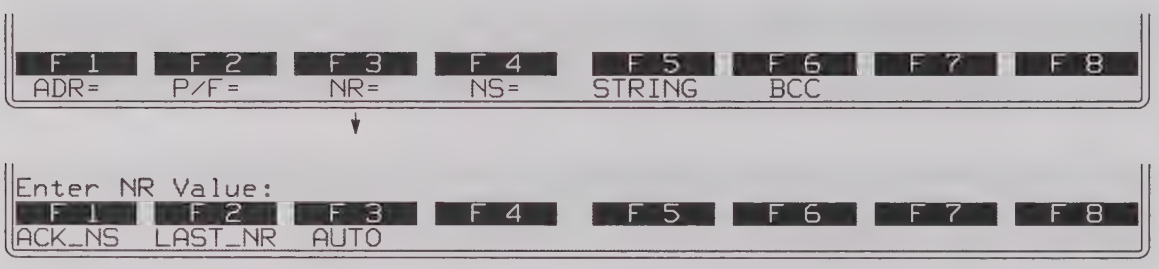


Figure 39-23 The N(R) field may be specified in INFO and supervisory frames to be sent.

8. *N(S)*. N(S) fields are transmitted in INFO frames only. (See the frame–field diagrams in Figure 39-5.) Entries for N(S) in SEND INFO actions are optional. The softkeys that open below NS= are illustrated in Figure 39-24.

   To specify an N(S) value, press the softkey for NS=, then enter a hexadecimal in the form of one or two alphanumerics. Valid hex entries are the same as for N(R). A SEND INFO action that *specifies* an N(S) value—NS= 0, for example—will clear the window so that the INFO frame is passed immediately to Layer 1.

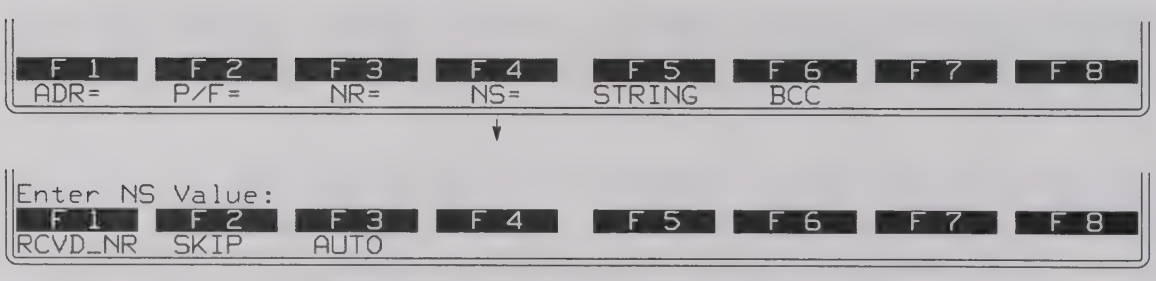


Figure 39-24 The N(S) field may be specified in a *SEND INFO* action.

Other N(S) options are RCVD_NR, SKIP, and AUTO. RCVD_NR means that you send the N(S) value that the other side says it is expecting. This is the valid N(S) in

many cases, but not when you send two or more I–frames in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct N(S). This will look to the other side as though the line has taken a "hit" and a frame has been lost. This selection causes the window to be cleared.

NS= AUTO is the default setting for SEND INFO actions. AUTO means that every *new* INFO frame sent will have an N(S) value of one higher than the previous N(S) to the same frame address.

9. *String*. Strings are sent at Layer 2 only as adjuncts to frame–types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

   Press the SEND softkey followed by the softkey for a frame type. Add any necessary or desired SEND options for the particular frame type. Then press the STRING softkey (Figure 39-24).

   There is no spreadsheet keyword that identifies send–strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " –character in your string, type \" .)

   Here is a simple SEND action that includes no options besides a string:

   ACTIONS: SEND FRMR "$^{1}{}_{1}{}^{6}{}_{6}{}^{0}{}_{8}$"

   And here is a SEND action that includes a full complement of optional fields, including a string:

   ACTIONS: SEND INFO ADR= C3 P/F= 0 NR= AUTO NS= AUTO
   "$^{2}{}_{C}{}^{0}{}_{0}{}^{0}{}_{2}{}^{0}{}_{0}{}^{0}{}_{0}{}^{0}{}_{0}{}^{0}{}_{3}{}^{8}{}_{0}{}^{0}{}_{0}{}^{1}{}_{5}$This is user data.$^{1}{}_{5}$" GDBCC

   Most ASCII–keyboard, control, and hexadecimal characters are legal in a send–string. Special keys (FLAG, DON'T CARE, BIT MASK) are not legal. Refer to Table 33-2.

   To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)) . Remember that the double parens are special characters produced by the CTRL–9 and CTRL–0 combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

10. *BCC*. There are three BCC options for every SEND action at Layer 2 SDLC. One of the options, GDBCC, is the default. Any frame that does not request a bad BCC or an abort will have a good frame–check sequence calculated for it and appended to it. BCC also is an option for SEND actions at Layer 1; but it does not occur at Layer 3 or higher.

    The three softkey selections for BCC are shown in Figure 39-25. A 16–bit CCITT frame check is selected automatically for BOP protocols and cannot be changed or disabled. A bad BCC will be CRC–16 instead of CCITT.

When ABORT is the BCC selection, instead of appending a proper frame check, the transmitter will hold the lead at mark for eight bits (or longer if the transmitter is idling[F]). Inside a frame, seven 1−bits in a row are sufficient to signal an abort.

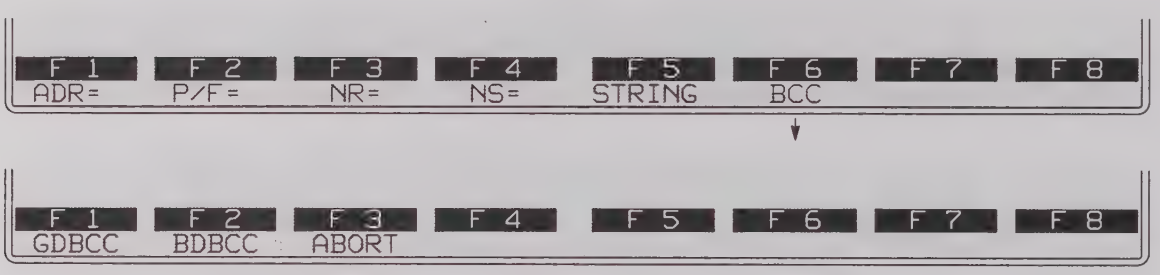


Figure 39-25 Type of BCC is a *SEND* option for frames at Layer 2.

(B) Give Data

GIVE_DATA is the F2 action on the first rack of action softkeys (refer to Figure 39-18). This action takes the I−field from a received INFO frame and passes it up to Layer 3 along with a DL_DATA IND primitive. (See Figure 34-5 in the section, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data is delivered up to Layer 3 only by one of two actions at Layer 2: GIVE_DATA, or else a DL_DATA IND primitive followed by the data string.

(C) Resend

The RESEND function is mapped to F1 on the second layer of action softkeys at Layer 2 for SDLC. See Figure 39-26. A RESEND action will resend the first frame in the window. The window is a queue that buffers INFO frames for retransmission in case one or more transmissions are lost or in error.

The first frame in the window always is the *earliest outstanding* (unacknowledged) frame. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first−frame" position is established. The first RESEND after an acknowledgment always sends the first window frame.

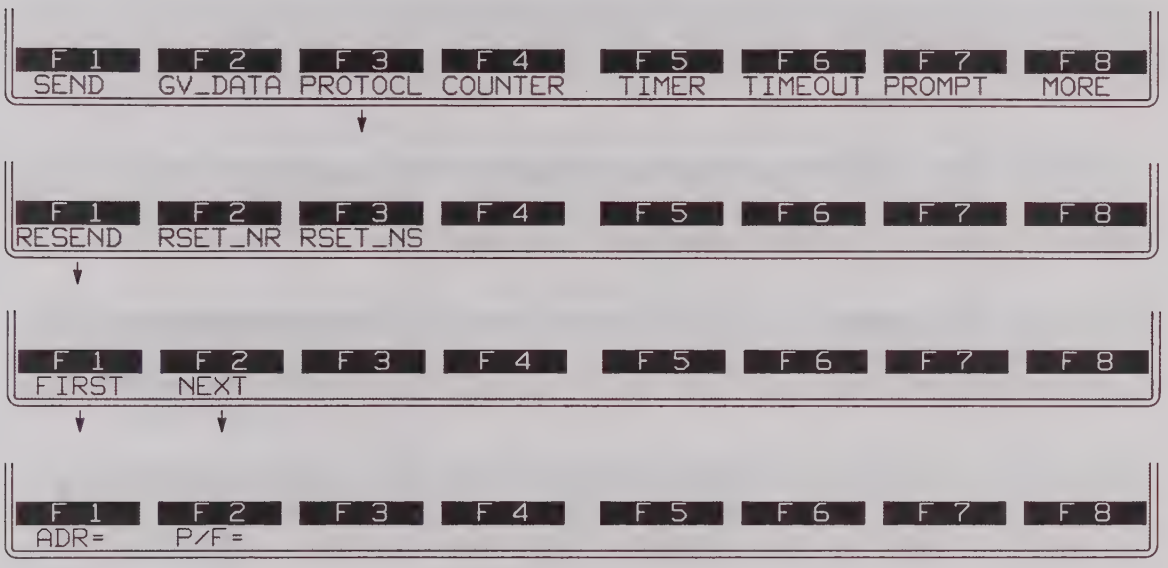


Figure 39-26 The *RESEND* action allows you to recover from sequence errors.

The *second and subsequent* RESENDs following an acknowledgment also will send the first window frame, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window frames. Figure 39-27 shows the position of the the resend "pointer" after four consecutive RESEND actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

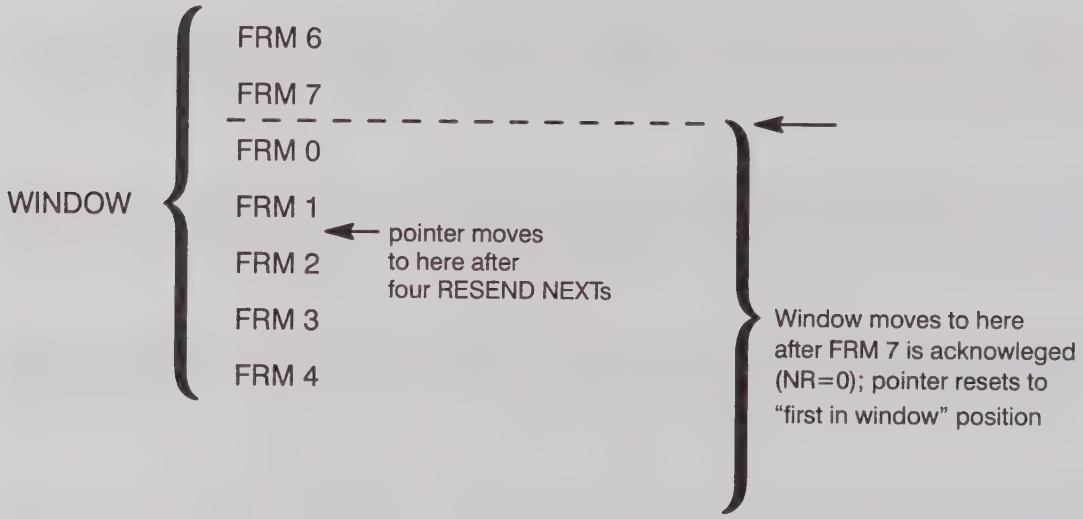


Figure 39-27 Resends cause the pointer to move, while acknowledgments move the pointer *and the entire window.*

The resend–pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

1. *Resend first/next.* RESEND FIRST means that the resend–pointer is reset to the beginning of the window, the first frame in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 39-28.

   The RESEND FIRST action makes it possible for you to resend all the frames in the window one by one, and then *resend them again* if necessary.

2. *ADR=* . Although not strictly required, you should designate the controller address of the resend frame. Two–digit hexadecimal values in the range 00 through FF are valid.

   There is also a LOOPBAK softkey selection for the address field. This selection causes the RESEND action to refer to the address of the most recently received frame.

   If you selected **Emulation Addressing:** MULTI–DROP on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 39.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can resend frames to any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will resend the appropriate frame from that address' transmit window. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not resend to any address.

When **Emulation Addressing:** POINT-TO-POINT is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

3. *P/F=*. The P/F−bit in the resend−frame can be set to zero or one by this optional action. (Default is 1 in a RESEND action.)

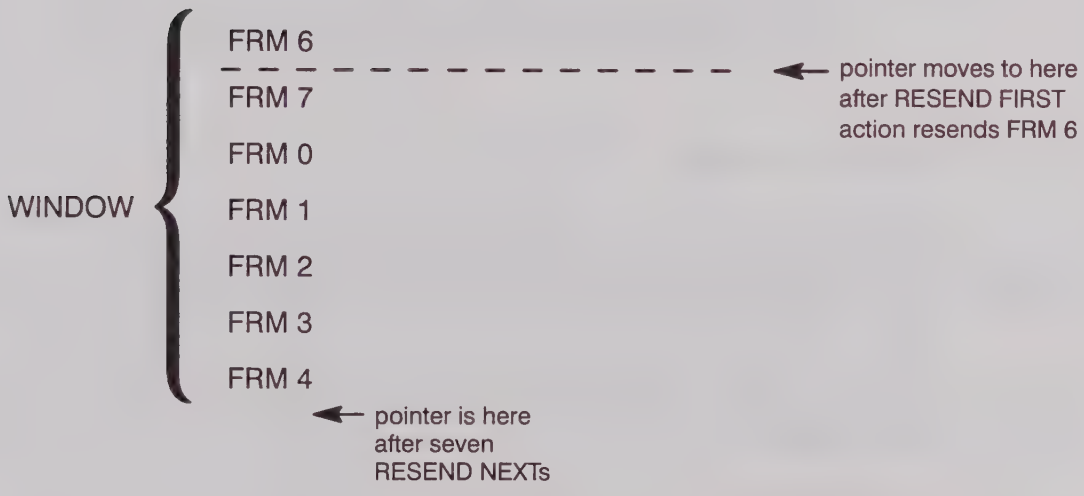


Figure 39-28 *RESEND FIRST* resets the pointer, allowing you to resend the entire window *repeatedly*.

(D) Reset N(R) and Reset N(S)

RESET_NR and RESET_NS are the F2 and F3 actions on the second rack of action softkeys for SDLC. (Refer to Figure 39-26.) The sequence−number fields in I−frames and supervisory frames can be reset by these two Protocol Spreadsheet actions. Sequence numbers are not reset *automatically* during a test by any frame that is sent or received.

RESET_NS also clears the transmit window.

If you press RSET_NR or RSET_NS, another softkey rack will be presented.

1. *ADR=*. Although not strictly required, you should designate a specific controller address. Two−digit hexadecimal values in the range 00 through FF are valid.

   There is also a LOOPBAK softkey selection for the address field. This selection causes the RESET_NR and RESET_NS actions to refer to the address of the most recently received frame.

If you selected **Emulation Addressing:** MULTI–DROP on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 39.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can reset N(R) or N(S) for any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will reset N(R)—or N(S)—for that address only. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not reset N(R)—or N(S)—for any address.

When **Emulation Addressing:** POINT–TO–POINT is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

39.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 2 protocol trace (see Section 39.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the SDLC protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F2 on the third rack of action softkeys. Figure 39-29 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

Figure 39-30 shows one screen of a Layer 2 protocol trace in which DTE SNRM frames have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

```
CONDITIONS: DTE SNRM
ACTIONS: ENHANCE REVERSE
```

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 39-29 shows the softkey path to SUPPRES.

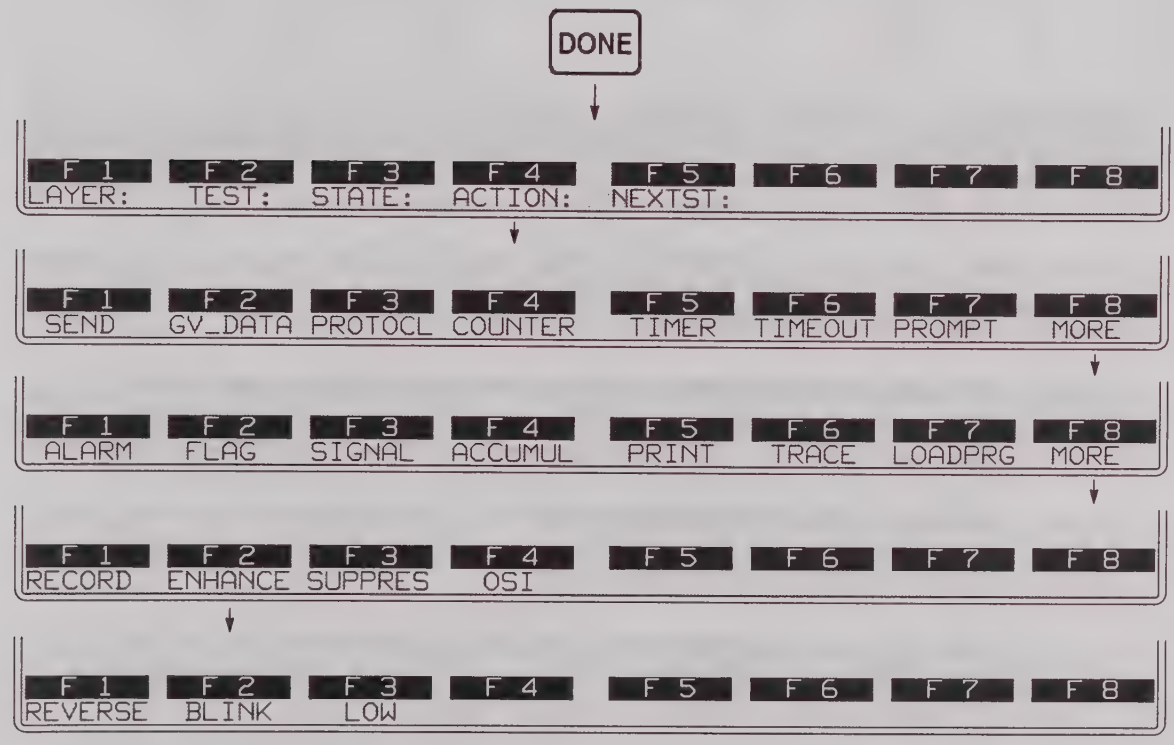


Figure 39-29 Selected frames on the protocol trace may be enhanced (or suppressed).

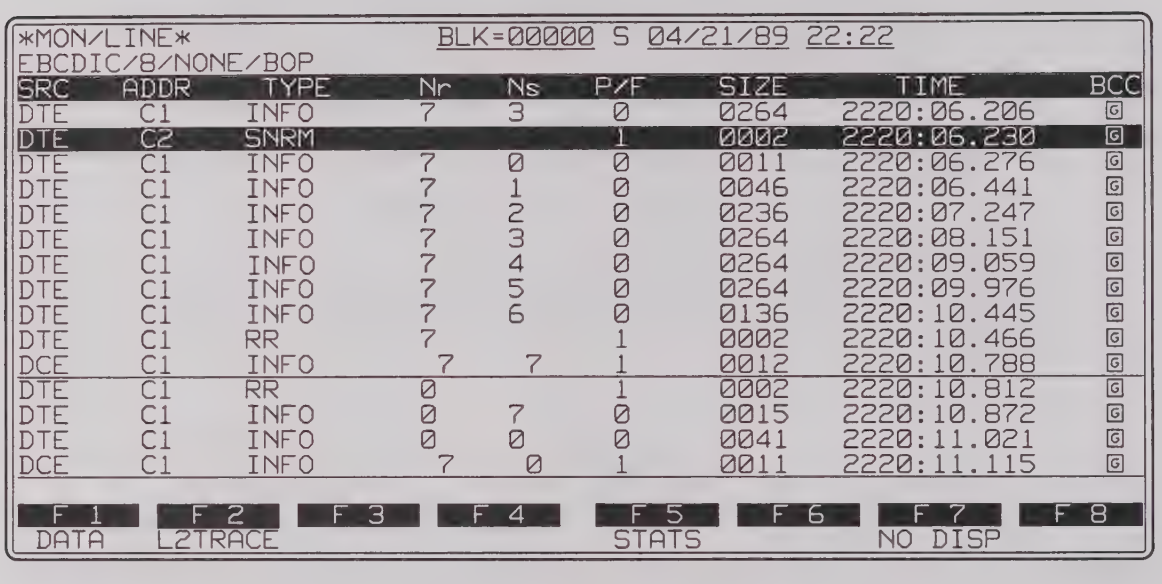


Figure 39-30 Set Normal Response Mode (SNRM) commands have been enhanced on the DTE side.

39.7 Automatic Primitives

In Section 34, Table 34-2 lists the OSI service primitives that are monitored at the boundaries of Layer 2 as trigger conditions and sent up to Layer 3, or down to Layer 1, as user-entered spreadsheet actions. These primitives are layer-specific rather than protocol-specific and are not part of the personality package for SDLC; but a few of the primitives are set in motion automatically by SDLC spreadsheet actions at Layer 2. These *automatic* primitives can be thought of as part of the Layer 2 actions themselves, and by extension, as part of the SDLC protocol package.

Table 39-1 gives the set of SDLC actions that have action-primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 2, a DL_DATA IND primitive is forwarded to Layer 3, where a DL_DATA IND condition may be waiting to monitor it.

Whenever a SEND or RESEND action is initiated at Layer 2, a PH_DATA REQ primitive is sent downward along with the PH data (the entire frame).

If a SEND or RESEND action is triggered at Layer 2 while the physical connection at Layer 1 is inactive, Layer 2 will sense the absence of a physical connection and delay the PH_DATA REQ. Instead, it will send a PH_ACTIVATE REQ primitive. Only when a PH_ACTIVATE CONF has been returned by Layer 1 will Layer 2 release the data and the data primitive.

NOTE: The RS−232 interface does not distinguish active/inactive status at the physical level. This interface returns PH_ACTIVATE CONF automatically whenever it sees PH_ACTIVATE REQ.

Table 39-1
Automatic Primitives Generated at Layer 2 SDLC

SDLC Layer 2 Action	Automatic Primitive	To Layer
GIVE_DATA	DL_DATA IND	3
SEND {TYPE}	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1
RESEND	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1

* Sent if Layer 1 shows inactive status. PH_DATA REQ delayed until PH_ACTIVATE CONF returned by Layer 1.

40 SNA

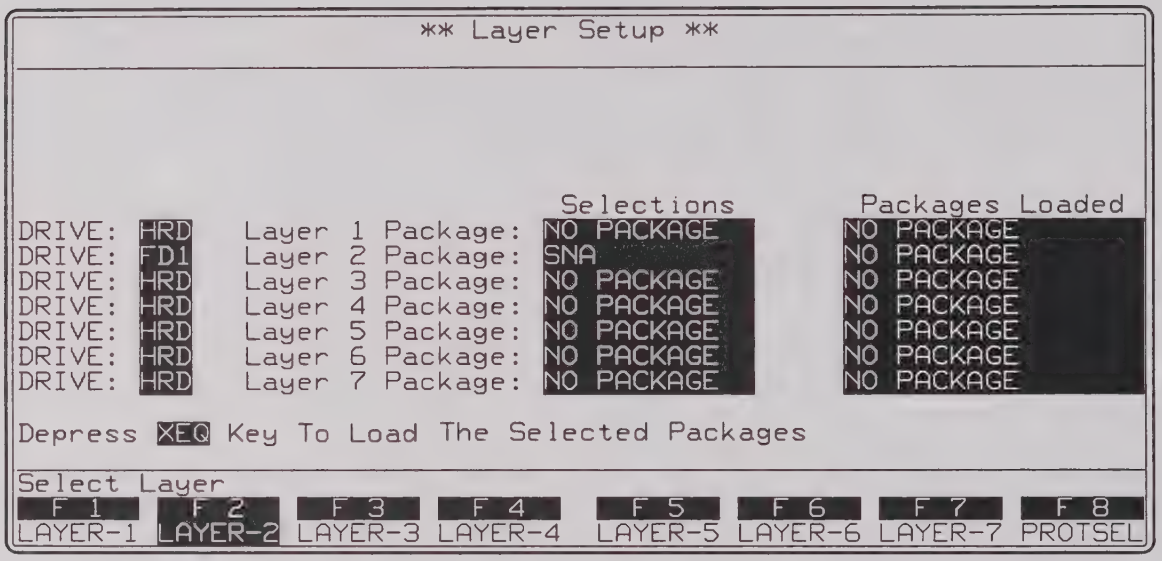


Figure 40-1 The *SNA* personality package for Layer 2 is loaded from the Layer Setup screen.

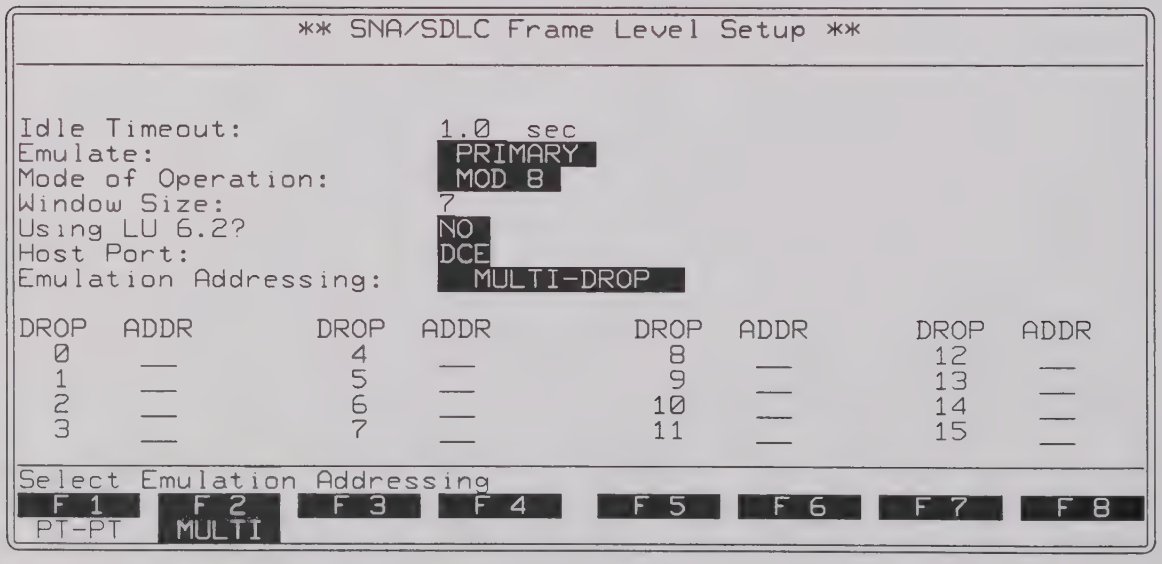


Figure 40-2 Protocol Configuration screen for SNA.

40 SNA

SNA is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 40-1 shows the Layer Setup screen configured to load in the SNA package from floppy−disk, Drive 1. Refer to Section 8 for details on operating the Layer Setup screen.

The SNA package consists of the following:

- A special SNA Frame Level Setup screen that controls certain parameters when the unit is emulating SDLC or tracing SNA and SDLC.
- Multidrop or point−to−point operation.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate SDLC programming. *This is the same set of conditions and actions that is documented in Section 39 of this manual.* Refer to Section 39 for a discussion of the spreadsheet conditions and actions that are specific to the SDLC and SNA Layer−2 packages.
- A protocol trace (illustrated in Figure 40-3) that distills from SNA data the SDLC and SNA events that have protocol significance. This trace is accessible by softkey in Run mode at all times.

40.1 Frame−Level Setup

The parameters on the SNA Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation. Use this screen also to enable multidrop operation.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the SNA selection at Layer 2: **SNA** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up a prompt to **Select Protocol Configuration Screen**. Then press F2 (LAYER−2) to call up the SNA Frame Level Setup screen.

The parameter fields on this screen are shown in Figure 40-2. **Idle Timeout**, **Emulate**, **Mode of Operation**, **Window Size**, **Emulation Addressing**, and **ADDR** are covered in Section 39.1 and will not be discussed here.

Using LU 6.2? allows you to set up the SNA trace to monitor LU 6.2 sessions correctly. The default selection in this field is NO .

Selection of DTE or DCE in the **Host Port** field allows the SNA protocol trace to distinguish properly between two kinds of FMD headers—NS headers and FM headers—in SNA Request Units.

40.2 SDLC Conditions and Actions

The same set of special conditions and actions documented for the SDLC Layer 2 package is implemented on the Protocol Spreadsheet in SNA. Refer to subsections 39.3 through 39.7.

40.3 Protocol Trace

The SNA package includes an automatic real–time trace that summarizes link–level, TH, RH, and RU activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE (F2 on the primary rack of display–mode softkeys) to bring the protocol trace for SNA to the screen. Figure 40-3 is an example of this trace display. Each DTE or DCE entry on the SNA trace marks the beginning of a frame.

In both Run and Freeze modes, the trace may be displayed as either hexadecimal data or text code data. The text code will be that selected on the Line Setup menu. Press HEX to toggle between the two display formats. You may print the Protocol Trace from Freeze mode in either hexadecimal or text code format. (See also Section 16.)

When running in High–Speed Frame Mode, more data could be passed to Layer 2 than there is room for in the buffer; this will cause an FEB overrun. If this happens, the error message **FE Buffer Overflowed – Some Frames Lost** will appear on the prompt line. The first time an FEB overrun occurs, an audible alarm will also sound; subsequent reccurrences will cause only the message to display (without any alarm). The trace will restart again but some data is lost with each occurrence.

(A) The Protocol Trace in Freeze Mode

Press FREEZE to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing ROLL FWD or ↓ moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing ROLL BACK or ↑ moves the viewing window up to add a line of older data to the top of the screen.

Depression of the NEXT PAGE key adds 16 lines—one full page—of newer trace data to the frozen trace screen. Depression of PREV PAGE adds 16 lines of older data.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys.

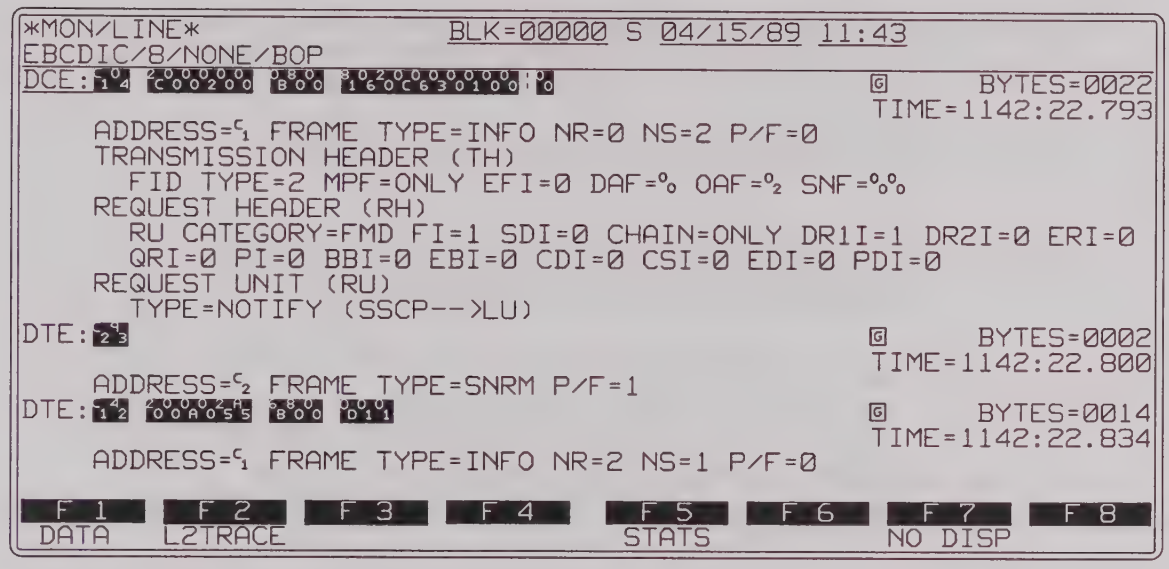


Figure 40-3 Each *DTE* or *DCE* entry on the SNA trace marks the beginning of a frame.

(B) Trace Fields Common to All Frames

1. *Source*. On the initial line in each frame expansion, the source of the frame (DTE or DCE) is identified. This is the only field on the frame–trace that begins at the left border of the screen, and it acts as a visible separator of frame expansions that take up varying amounts of display space. Note on the screen in Figure 40-3 that two whole frame expansions and part of a third are presented.

   In the leftmost ("Source") column in the trace, DCE is always underlined.

2. *Character data*. Character data decoded in hexadecimal is presented in reverse video immediately following the source field. See Figure 40-3. Character data is presented in blocks separated by spaces. These blocks correspond to the following protocol fields:

   - Frame header
   - Transmission header (TH)
   - Request/response header (RH)
   - Request/response unit (RU)

When an individual block is longer than ten bytes, another separator is used: the vertical bar (¦). Bars occur after every ten bytes within a block. Note the bar in the RU block of the DCE frame at the top of Figure 40-3.

The hex–character display does not exceed two lines on the trace, or 70 bytes of RU.

Note that the four protocol blocks may not be present in every unit. Non–Info frames have only the frame header. Middle– and last–segment messages lack the RH block. Many response messages lack the RU.

3. *Frame checking.* An SDLC frame ends as soon as a $^{7}_{E}$ flag or seven 1–bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the BCC field on the first trace line for the frame. The symbol G denotes a good frame check, while B symbolizes a bad frame. A for abort is posted to the displays when a frame is ended by seven 1–bits.

4. *Size.* The number of bytes in each frame is given in the BYTES= field in four decimal digits. The count begins with the address byte and excludes the two–byte FCS. Frames without I–fields show a byte–count of two.

5. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

6. *Address*. The address byte is given as a hexadecimal character (from $^{0}_{0}$ to $^{F}_{F}$) in the ADDRESS= field.

7. *Frame type*. The mnemonic (abbreviated) name for the frame type is given in the FRAME TYPE= field.

8. *P/F*. The status of the poll or the final bit is given in the P/F= field.

(C) Other Trace Fields

Most of the fields on the SNA trace are not common to all frames, but are specific to the type of frame (Info, for example), FID type, request messages, response messages, RU type, and so forth. These fields are included in Table 40-1.

Table 40-1
Fields in SNA Trace Display

Mnemonic	Name	Data Columns # cols in field	Data Columns value each column	Meaning
Frame header				
(SOURCE=)			ASCII	DTE, DCE
(BCC=)		1	symbol	G, B, A
BYTES=		4	0–9	
TIME=		9	0–9	time of end of frame, given as hhmm:ss.mmm
ADDRESS=		1	$^0_0$ –^F_F	
FRAME TYPE=			ASCII	SNRM, INFO, etc.
NR= (mod 8)	number (next) receive frame	1	0–F	present only if frame type=info, rr, rnr, rej, or srej
NR= (mod 128)	number (next) receive frame	2	0–F	present only if frame type=info, rr, rnr, rej, or srej
NS= (mod 8)	number (frame) sent	1	0–F	present only if frame type=info
NS= (mod 128)	number (frame) sent	2	0–F	present only if frame type=info
P/F=	poll/final	1	0–1	
Transmission header, FID 0 or 1				
FID TYPE=0, 1	format identifier			10–byte th
MPF=	mapping field		ASCII	MIDDL, LAST, FIRST, ONLY
EFI=	expedited flow indicator	1	0–1	0=normal 1=expedited
DAF=	destination address field	2	$^0_0$ –^F_F	
OAF=	origin address field	2	$^0_0$ –^F_F	
SNF=	sequence number field	2	$^0_0$ –^F_F	
DCF=	data count field	2	$^0_0$ –^F_F	
Transmission header, FID 2				
FID TYPE=2	format identifier			6–byte th
MPF=	mapping field		ASCII	MIDDL, LAST, FIRST, ONLY
EFI=	expedited flow indicator	1	0–1	0=normal 1=expedited
DAF=	destination address field	1	$^0_0$ –^F_F	
OAF=	origin address field	1	$^0_0$ –^F_F	
SNF=	sequence number field	2	$^0_0$ –^F_F	
Transmission header, FID 3				
FID TYPE=3	format identifier			2–byte th
MPF=	mapping field		ASCII	MIDDL, LAST, FIRST, ONLY
EFI=	expedited flow indicator	1	0–1	0=normal 1=expedited
SESSION=			ASCII	SSCP–PU, SSCP–LU, RESERVD, LU–LU
LAF=	local address field	6	0–1	

Table 40-1(continued)
Fields in SNA Trace Display

Mnemonic	Name	Data Columns		Meaning
		# cols in field	*value each column*	
Transmission header, FID 4				
FID TYPE=4	format identifier			26–byte th
TGSI=	transmission group sweep indicator	1	0–1	1=piu order maintained in tg
VRSI=	er & vr support indicator	1	0–1	1=one or more nodes does not support er, vr
VRPCI=	virtual route pacing count indicator	1	0–1	0=vr pacing count not equal to zero
NP=	network priority	1	0–1	1=piu flows at network priority, not tpf
IERN=	initial explicit route number	1	0–F	same as vrn
ERN=	explicit route number	1	0–F	ern + vrn + tpf = vrid
VRN=	virtual route number	1	0–F	
TPF=	transmission priority field	1	0–3	0=low 1=medium 2=high
VRCWI=	vr change window indicator	1	0–1	0=increment 1=decrement
TGNFI=	tg non–fifo indicator	1	0–1	0=tg fifo required
VRSTI=	vr sequence & type indicator	1	0–3	0=nonseq, nonsupv 1=nonseq, supv 2=singly seq
TGSNF=	tg sequence number field	3	0–F	used when tgnfi=0
VRPRQ=	virtual route pacing request	1	0–1	1=vr pacing response requested
VRPRS=	virtual route pacing response	1	0–1	1=vr pacing response sent in response to vrprq=1
VRCWRI=	vr change window reply indicator	1	0–1	when vrprs=1, 0=increment 1=decrement
VRRWI=	virtual route reset window indicator	1	0–1	0=do not reset 1=reset window to minimum size
VRSSN=	virtual route send sequence number	3	0–F	piu sequencing for vrsti=2
DSAF=	destination subarea address field	4	$^{0}_{0}$ –$^{F}_{F}$	dsaf + osaf + vrn = vr
OSAF=	origin subarea address field	4	$^{0}_{0}$ –$^{F}_{F}$	
SNAI=	sna indicator	1	0–1	0=destination is non–sna device, convert to fid 0
MPF=	mapping field		ASCII	MIDDL, LAST, FIRST, ONLY
EFI=	expedited flow indicator	1	0–1	0=normal 1=expedited
DEF=	destination element field	2	$^{0}_{0}$ –$^{F}_{F}$	dsaf + def = destination network address
OEF=	origin element field	2	$^{0}_{0}$ –$^{F}_{F}$	osaf + oef = origin network address
SNF=	sequence number field	2	$^{0}_{0}$ –$^{F}_{F}$	biu sequence number, segments have same snf
DCF=	data count field	2	$^{0}_{0}$ –$^{F}_{F}$	biu or biu segment length for piu blocking
Transmission header, FID F				
FID TYPE=F	format identifier			26–byte th
CF=	command format	1	$^{0}_{0}$ –$^{F}_{F}$	$^{0}_{1}$
CT=	command type	1	$^{0}_{0}$ –$^{F}_{F}$	$^{0}_{1}$=tg snf wrap acknowledgment
CSN=	command sequence number	2	$^{0}_{0}$ –$^{F}_{F}$	
DCN=	data count number	2	$^{0}_{0}$ –$^{F}_{F}$	

Table 40-1 (continued)
Fields in SNA Trace Display

Mnemonic	Name	Data Columns # cols in field	Data Columns value each column	Meaning
Request header				
RU CATEGORY=		ASCII	FMD, NC, DFC, SC	
FI=	format indicator	1	0–1	ru category=fmd, lu–lu session: 1=fm header follows ru category=fmd, sscp session: 0=character-coded ru 1=field-formatted ru ru category=nc, dfc, sc: fi always = 1
SDI=	sense data indicator	1	0–1	1=sense data included
CHAIN=			ASCII	MIDDL, LAST, FIRST, ONLY
DR1I=	definite response 1 indicator	1	0–1	0=no response requested (but may be requested by dr2i) 1=response requested
DR2I=	definite response 2 indicator	1	0–1	0=no response requested (but may be requested by dr1i) 1=response requested
ERI=	exception response indicator	1	0–1	if response requested by dr1I or dr2i 0=definite response 1=exception response
QRI=	queued response indicator	1	0–1	0=bypass tc queues
PI=	pacing indicator	1	0–1	1=pacing request
BBI=	begin bracket indicator	1	0–1	1=begin bracket
EBI=	end bracket indicator	1	0–1	configured for non–LU 6.2 (see Sec. 40.1), 1=end bracket
CDI=	change direction indicator	1	0–1	1=change direction
CSI=	code selection indicator	1	0–1	0=code 0 1=code 1
EDI=	enciphered data indicator	1	0–1	1=ru is enciphered
PDI=	padded data indicator	1	0–1	1=ru was padded before encipherment
CEBI=	conditional end bracket indicator	1	0–1	configured for lu 6.2, 1=conditional end bracket
Response header				
RU CATEGORY=		ASCII	FMD, NC, DFC, SC	
FI=	format indicator	1	0–1	ru category=fmd, lu–lu session: 1=fm header follows ru category=fmd, sscp session: 0=character-coded ru 1=field-formatted ru ru category=nc, dfc, sc: fi always = 1
SDI=	sense data indicator	1	0–1	1=sense data included
CHAIN=			ASCII	ONLY
DR1I=	definite response 1 indicator	1	0–1	1=dr1i response
DR2I=	definite response 2 indicator	1	0–1	1=dr2i response
RTI=	response type indicator	1	0–1	0=positive response 1=negative response
QRI=	queued response indicator	1	0–1	0=response bypasses tc queues 1=enqueue response in tc queues
PI=	pacing indicator	1	0–1	1=pacing response

Table 40-1(continued)
Fields in SNA Trace Display

Mnemonic	Name	Data Columns # cols in field	Data Columns value each column	Meaning
Request unit				
(REQUEST CODE=)			ASCII	ACTLU, BIND, etc. (present if ru category=nc, dfc, or sc)
(FMD NS HEADER=)			ASCII	CONTACT, NOTIFY, etc. (sscp session only: present if ru category=fmd and fi=1)
(FM HEADER) TYPE=		1	0–9	lu–lu session only: present if ru category=fmd and fi=1
(BIND TYPE=)			ASCII	NEGOTIABLE, NONNEGOTIABLE (this field and remaining request–unit fields present only if request code=bind)
FM PROFILE =		1	$^{0}_{0}$ –$^{F}_{F}$	
TS PROFILE=		1	$^{0}_{0}$ –$^{F}_{F}$	
FM USAGE PRIMARY LU PROTOCOLS FOR FM DATA=		1	$^{0}_{0}$ –$^{F}_{F}$	
FM USAGE SECONDARY LU PROTOCOLS FOR FM DATA=		1	$^{0}_{0}$ –$^{F}_{F}$	
FM USAGE COMMON LU PROTOCOLS=		2	$^{0}_{0}$ –$^{F}_{F}$	
TS USAGE=		6	$^{0}_{0}$ –$^{F}_{F}$	
MAX RU FROM SLU=		6	0–9, UNKNOWN	
MAX RU FROM PLU=		6	0–9, UNKNOWN	
PS PROFILE=		1	$^{0}_{0}$ –$^{F}_{F}$	
PS CHARACTERISTICS		11	$^{0}_{0}$ –$^{F}_{F}$	
USER COUNT		1	$^{0}_{0}$ –$^{F}_{F}$	length of user data in bind ru
Response unit				
(REQUEST CODE=)			ASCII	ACTLU, BIND, etc. (present if ru category=nc, dfc, or sc)
(FMD NS HEADER=)			ASCII	ADDLINK, CDINIT, etc. (present if ru category=fmd and fi=1)
SENSE DATA=		4	$^{0}_{0}$ –$^{F}_{F}$	this and the following 3 fields present only if sdi=1
CATEGORY=			ASCII	USER SENSE DATA ONLY, REQUEST REJECT, REQUEST ERROR, STATE ERROR, RH USAGE ERROR, PATH ERROR
MODIFIER=		1	$^{0}_{0}$ –$^{F}_{F}$	
SENSE INFORMATION=		2	$^{0}_{0}$ –$^{F}_{F}$	

41 DDCMP Layer 1

DDCMP is a "layer personality package" of functions loaded into memory from disk via the Layer Setup screen. Figure 41-1 shows the Layer Setup screen with the DDCMP package loaded in from the hard−disk drive. Refer to Section 8 for information on operating the Layer Setup screen.

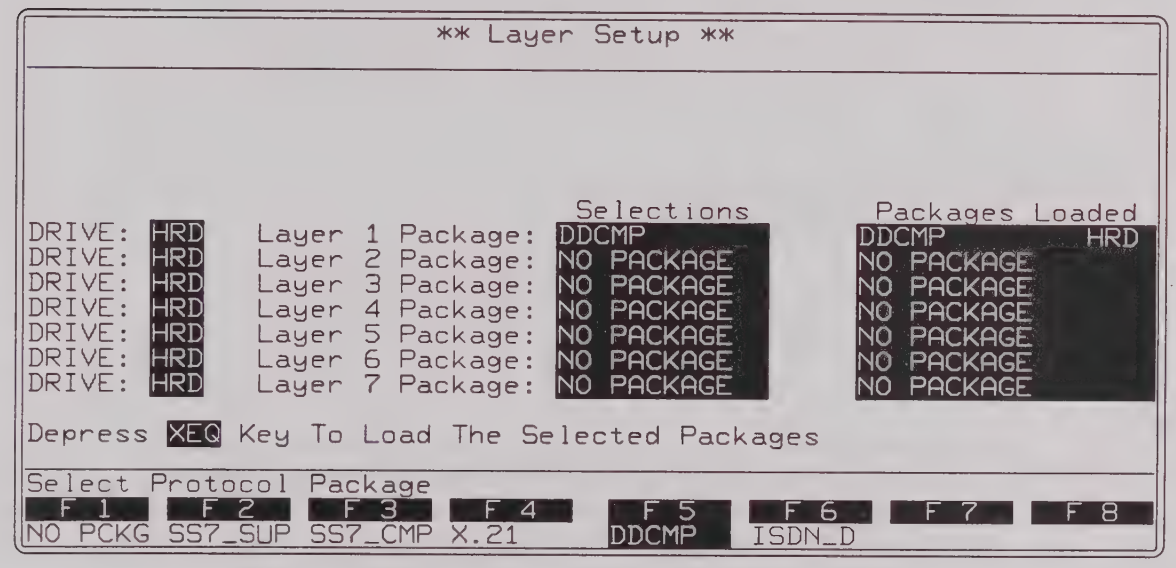


Figure 41-1 DDCMP is a "layer−personality package" of softkey functions at Layer 1.

The DDCMP package takes control of two functions that normally are configured by the user on the Line Setup menu: outsync and block checking. Control of these functions from the Line Setup menu is disabled when the DDCMP package is loaded in.

41.1 Outsync

In synchronous format, the sync pattern is selectable by the user in the Sync Char field on the Line Setup menu. Outsync parameters are not selectable. Outsync cannot be turned off. A receiver will go out of sync at the end of a message unless the first byte of the new message is ␖, ␗, or ␐ with the correct parity.

41.2 Block Checking

Screen display of good and bad BCCs is automatic when DDCMP is loaded in at Layer 1, and cannot be disabled on the Line Setup screen. The BCC setup for DDCMP cannot be modified or controlled in any way from the BCC Setup menu.

The results of both header and data block checks are displayed on the screen. If you want your program to detect good or bad BCCs, you may use the BCC selections on the trigger menus and at Layer 1 of the Protocol Spreadsheet to interrogate the header block check only.

If you want to detect a good or bad data block check, you must use one of the following C event variables:

```
extern fast_event fevar_gd_bcc2_td;
extern fast_event fevar_gd_bcc2_rd;
extern fast_event fevar_bd_bcc2_td;
extern fast_event fevar_bd_bcc2_rd;
```

Here is a program that counts bad DTE BCCs for both header and data:

```
{
   extern fast_event fevar_bd_bcc2_td;
}
LAYER: 1
   STATE: count_all_bad_dte_bccs
      CONDITIONS: DTE BAD_BCC
      ACTIONS: COUNTER t_bdbcc INC
      CONDITIONS:
      {
         fevar_bd_bcc2_td
      }
      ACTIONS: COUNTER t_bdbcc INC
```

Note to C programmers: the DDCMP Layer 1 package takes every message and places it in an IL buffer for use at Layer 2 and above. At the same time, it triggers the event *m_lo_ph_prmtv* and updates the variables associated with upward–going monitor–path primitives at Layer 2 (see Table 62-3). As a result, the OSI condition PH_TD_DATA IND (or PH_RD_DATA IND) comes true at Layer 2.

42 ISDN D Channel

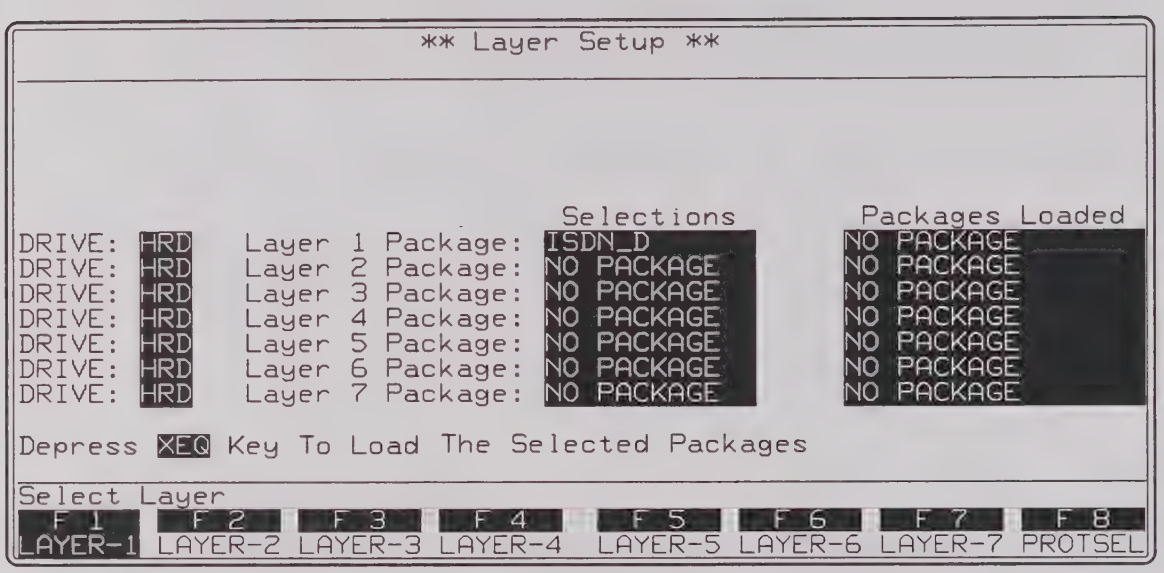


Figure 42-1 The ISDN_D personality package may be loaded from the Layer Setup screen.

42 ISDN D Channel

The Basic Rate ISDN service provides an aggregate data rate of 192 Kbps, with 144 Kbps available to users: two 64−Kbps B−channels and one 16−Kbps D−channel per interface (2B+D). The additional 48 Kbps are used for framing and maintenance. The INTERVIEW 8200 *TURBO* units, INTERVIEW 8600 *TURBO* units, INTERVIEW 8700 *TURBO* units, and INTERVIEW 8800 *TURBO* units, with an ISDN TIM and its multiplexer board in place (OPT−951−563−1 or OPT−951−566−1), support Basic Rate ISDN testing. Primary Rate ISDN data is carried over T1 and G.703 circuits.

> **NOTE:** The INTERVIEW 8100 *TURBO* unit supports neither Basic Rate ISDN testing nor Primary Rate ISDN testing.

ISDN_D is a "layer personality package" of functions loaded into memory from disk at Layer 1 via the Layer Setup screen. Figure 42-1 shows the Layer Setup screen configured to load in the ISDN_D package from the hard drive. Refer to Section 8 for information on operating the Layer Setup screen.

The ISDN_D package consists of a set of three C−language event variables and two C routines (see Section 75). These variables and routines allow the C programmer to construct Q.921 (LAPD) and Q.931 functions for use on the D channel. The ISDN trace application package (available as OPT−951−35) is built upon the D−channel variables and routines provided at Layer 1 by the ISDN_D package.

The ISDN_D layer package allows the user to send, receive, and monitor frames on the D channel via an application program written in C. Meanwhile, the line setup, data display, and layer packages can be focused on channel B1 or B2, whichever is selected on the ISDN Interface Setup menu (see accompanying documentation for the ISDN option). For Primary Rate ISDN in T1 or G.703, the ISDN_D layer package is also loaded at Layer 1 for the same use on the D channel as in Basic Rate ISDN; the B channel is selectable on their respective Interface Control screens.

> **NOTE:** The ISDN_D package should *not* be loaded when the D channel is selected for single−channel monitoring or emulating; that is, when the D channel is selected in the **Channel:** field on the ISDN Interface Setup menu.

43 LAPD

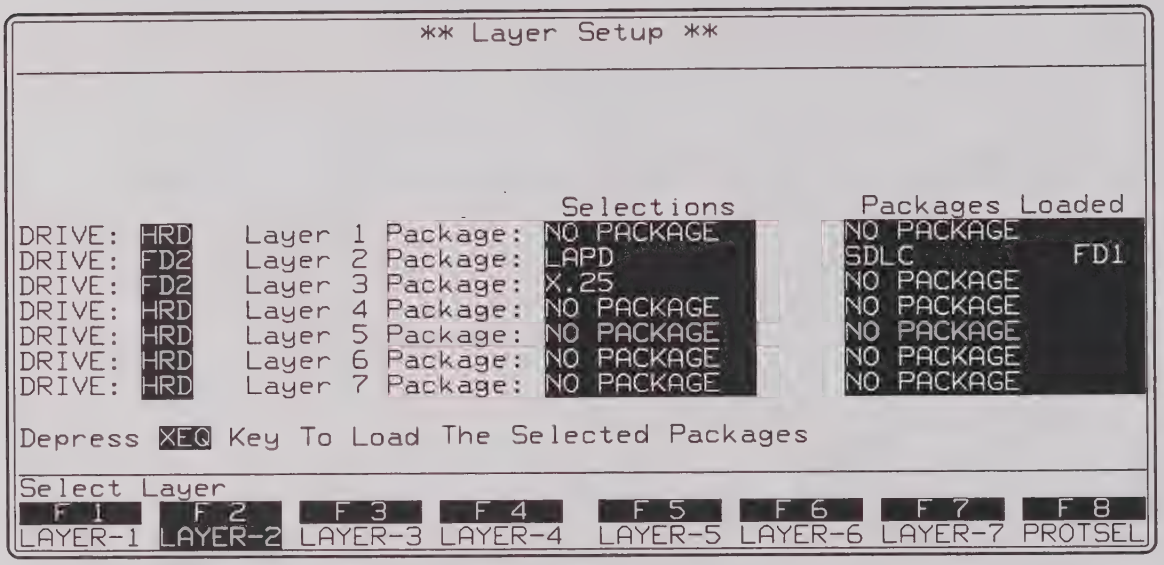


Figure 43-1 The LAPD personality package is loaded from the Layer Setup screen.

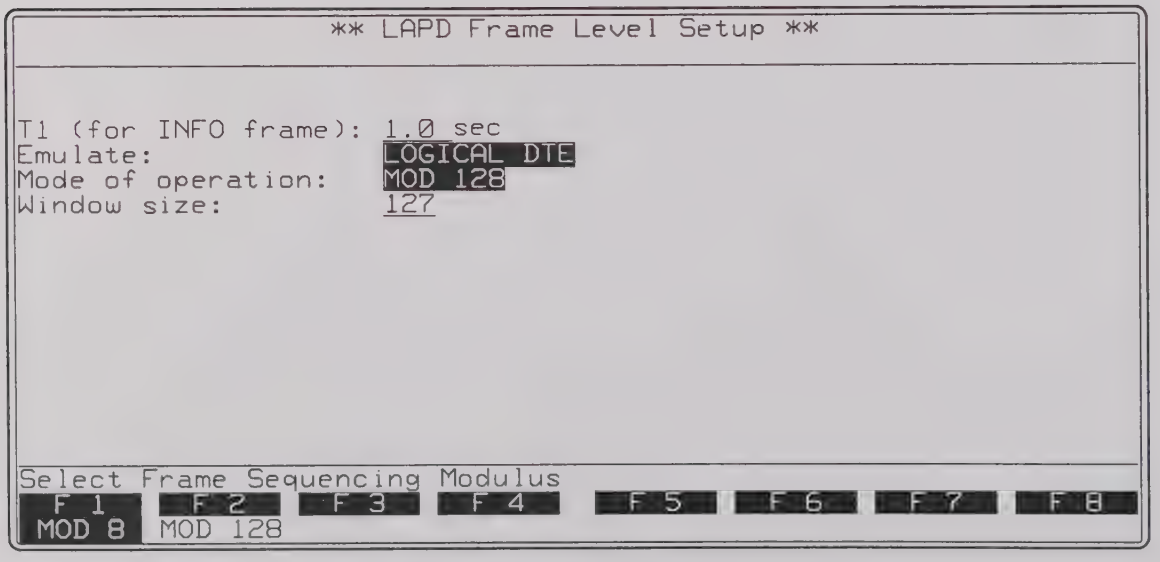


Figure 43-2 Protocol Configuration screen for LAPD.

43 LAPD

LAPD is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 43-1 shows the Layer Setup screen configured to load in the Layer 2 LAPD package from floppy−disk, Drive 2. Refer to Section 8 for details on operating the Layer Setup screen.

The LAPD package consists of the following:

- A special LAPD Frame Level Setup screen that controls certain parameters when the unit is tracing or emulating LAPD.
- A protocol trace (illustrated in Figure 43-3) that distills from LAPD data the Level 2 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate LAPD programming. Figure 43-5 shows the softkey path to the first rack of condition softkeys when the LAPD package is loaded in at Layer 2.

43.1 Frame−Level Setup

The parameters on the LAPD Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the LAPD selection at Layer 2: **LAPD** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up a prompt to **Select Protocol Configuration Screen**. Then press F2 (LAYER−2) to call up the LAPD Frame Level Setup screen.

The four parameter fields on this screen are shown in Figure 43-2. **T1**, **Emulate**, and **Window Size** apply to interactive (emulate) tests only. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) T1

Enter a four–digit (including decimal point) T1 timeout value in this field. The largest valid entry is 65.5 seconds. The smallest entry is .001 second, or 1 millisecond.

T1 is the name given to the retransmission timer for INFO frames. When a value is entered in the **T1** field on this menu, the Layer 2 package will handle T1 timings correctly, as follows:

- Whenever the INTERVIEW sends an I–frame at Layer 2 and *there are no previous frames sent by the INTERVIEW currently outstanding (unacknowledged)*, the timer starts timing down from the value entered on the Frame Level Setup screen.
- An acknowledgment by the device under test of the most recent frame transmitted by the INTERVIEW *stops* the timer (so that it does not expire).
- An acknowledgment by the device under test of a frame that is not the most recent frame transmitted by the INTERVIEW—an "incomplete" acknowledgment—restarts the T1 timer to the value selected on the configuration screen.

Expiration of this Frame Level Setup timeout can only be detected by a T1_EXPIRED condition on the Protocol Spreadsheet at Layer 2. This particular timeout cannot be detected by a generic condition of TIMEOUT T1.

According to the protocol, a T1_EXPIRED condition should result in a RESEND action.

(B) Emulate Logical DTE/DCE

There are two selections in the **Emulate** field on the LAPD Frame Level Setup screen, LOGICAL DTE and LOGICAL DCE. Usually, a logical DTE represents the user side of a link and a logical DCE is the network side of the link.

Use the **Mode** selection (EMULATE DTE or EMULATE DCE) on the Line Setup menu to regulate the *physical* interface—whether to use Pin 2 or Pin 3 to transmit, and so on.

(C) Mode of Operation

The **Mode of Operation** field refers to the mode of numbering INFO and supervisory frames. There are two options, MOD 8 and MOD 128.

MOD 8 uses sequence numbers 0–7. MOD 128 adds an extra byte to the control field in INFO, RR, RNR , and REJ frames. See Figure 43-4. This extra byte allows sequence numbers in a range of 0–127.

The correct "modulus" must be selected in this field in order to program successfully in Monitor mode and also to generate an accurate LAPD trace.

(D) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged I−frames that Layer 2 will buffer for retransmission. When the limit is reached, any further INFO frames that are named in SEND actions triggered at Layer 2 will be passed to Layer 1 for transmission but not buffered for retransmission.

The window is a queue that buffers frames for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) frame in the window. Successive RESENDs will send successive frames until there are no more frames to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

43.2 Protocol Trace

The LAPD package includes an automatic frame−trace display that summarizes link−level activity. This trace mode is enabled whenever the unit is in Run mode, both real−time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE to bring the protocol trace for LAPD to the screen. (If the Q.931 package for Layer 3 is also loaded in, the L2TRACE softkey will appear after you have pressed PROTOCL, F2 on the primary rack of display−mode softkeys.)

When running in High−Speed Frame Mode, more data could be passed to Layer 2 than there is room for in the buffer; this will cause an FEB overrun. If this happens, the error message **FE Buffer Overflowed − Some Frames Lost** will appear on the prompt line. The first time an FEB overrun occurs, an audible alarm will also sound; subsequent reccurrences will cause only the message to display (without any alarm). The trace will restart again but some data is lost with each occurrence.

Figure 43-3 is an example of the Layer 2 trace display. Each horizontal row in the trace represents a frame.

(A) The Protocol Trace in Freeze Mode

Press FREEZE to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing ROLL FWD or ↓ moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing ROLL BACK or ↑ moves the viewing window up to add a line of older data to the top of the screen.

Depression of the NEXT PAGE key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of PREV PAGE adds 15 lines of older frames.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys.

MON/LINE BLK=00000 S 06/23/89 09:15
ASCII/8/NONE/BOP

SRC	SAPI	TEI	C/R	TYPE	Nr	Ns	P/F	SIZE	TIME	BCC
DCE	04	1A	1	DISC			0	0003	0915:08.021	G
DTE	04	1A	1	UA			0	0003	0915:08.172	G
DTE	04	1A	0	SABM			1	0003	0915:08.212	G
DCE	04	1A	0	UA			1	0003	0915:08.440	G
DTE	04	1A	0	INFO	0	0	0	0007	0915:08.482	G
DCE	04	1A	1	INFO	0	0	0	0008	0915:08.723	G
DTE	04	1A	1	RR	1		0	0003	0915:08.842	G
DCE	04	1A	0	RR	1		0	0003	0915:08.915	G
DTE	04	1A	0	INFO	1	1	0	0019	0915:08.916	G
DCE	04	1A	0	RR	2		0	0003	0915:08.959	G
DCE	04	1A	1	INFO	2	1	0	0006	0915:09.033	G
DTE	04	1A	1	RR	2		0	0003	0915:09.362	G
DCE	04	1A	1	INFO	2	2	0	0018	0915:09.505	G
DTE	04	1A	1	RR	3		0	0003	0915:09.667	G
DTE	04	1A	0	INFO	3	2	0	0006	0915:09.747	G

F1 F2 F3 F4 F5 F6 F7 F8
L2TRACE L3TRACE

Figure 43-3 Each horizontal row on the trace display represents a frame.

(B) Trace Columns

The columns in the protocol trace for Layer 2 LAPD are explained below.

1. *Source.* The **SRC** column identifies the lead on which the frame was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the frame, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

   Just as on the data display, RD data is underlined.

2. *SAPI.* The Service Access Point Identifier (**SAPI**) is given in the next column. The SAPI is a network link–station address that appears in a six–bit field in the first frame–address byte: see Figure 43-4.

   The SAPI is presented on the trace display as two hex digits, with the righthand digit expressing the four low–order bits of the SAPI and the lefthand digit expressing the two remaining bits.

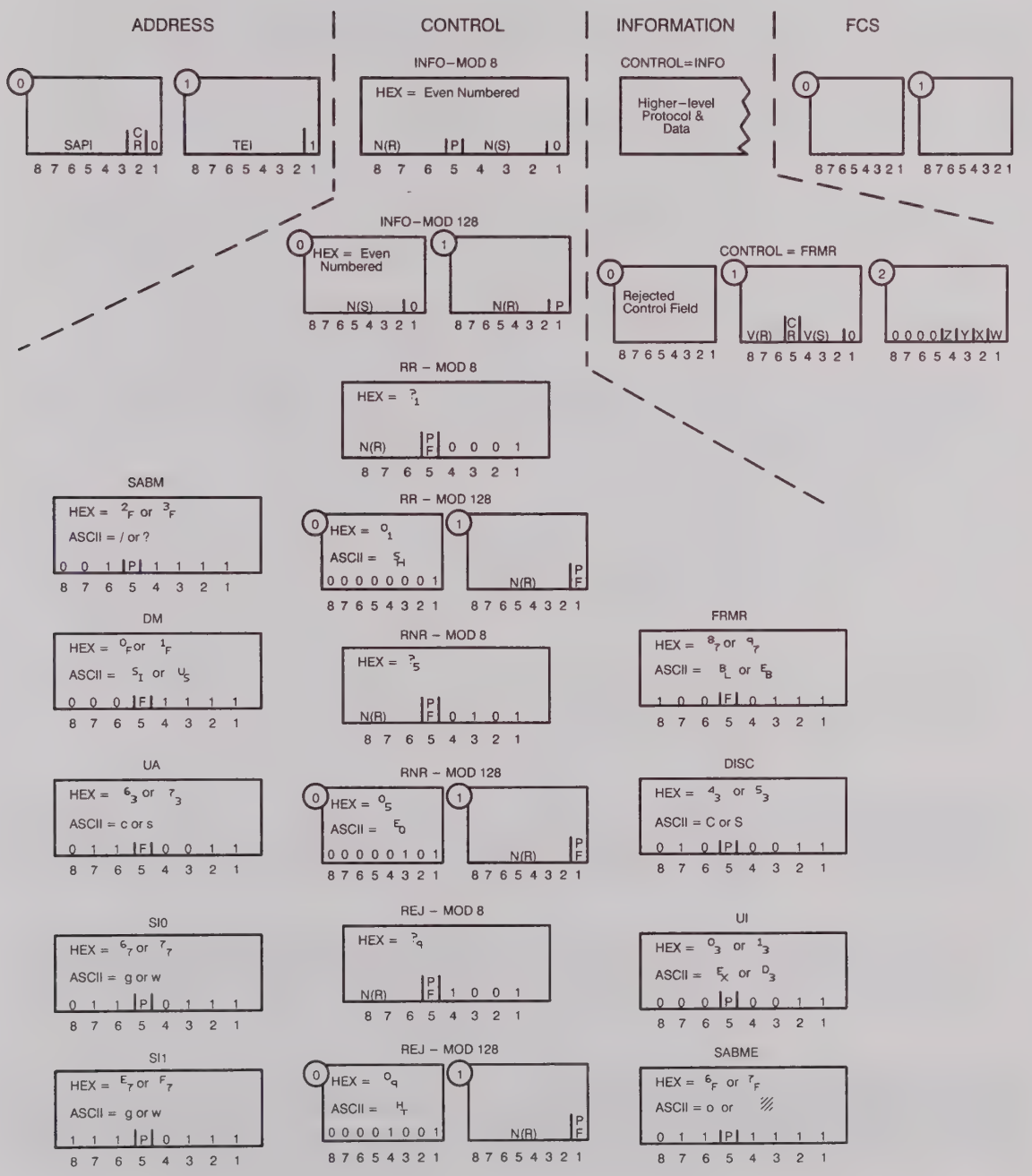


Figure 43-4 Frame fields in LAPD.

3. *TEI.* The Terminal Endpoint Identifier (**TEI**) is the address of a link station on the user side. It occupies seven bits in the second frame byte inside the leading flag (see Figure 43-4).

   The TEI is presented on the trace display as two hex digits, with the righthand digit expressing the four low–order bits of the TEI and the lefthand digit expressing the three remaining bits.

4. *C/R.* The Command/Response (**C/R**) column identifies the logical DTE (user side) and the logical DCE (network side). The logical DTE uses C/R 0 for INFO frames and other command frames, and C/R 1 for responses. The logical DCE uses C/R 1 for INFO frames and other commands, and C/R 0 for responses.

5. *Type.* The mnemonic (abbreviated) names for thirteen frame types as they appear in the **TYPE** column of the protocol trace are shown in Figure 43-4 under "CONTROL." The control field, therefore, indicates the frame type. If a control octet does not fit any of the patterns in the figure, the frame is listed in the **TYPE** column as UNKWN followed by the hexadecimal value of the control byte: UNKWN=47.

   If the number of bytes in the frame is below the required minimum, the frame is posted as INVALID.

6. *N(R) and N(S).* One column on the frame–level trace is devoted to N(R) values, and one column to N(S). The frame types that include N(R) or N(S) fields in their control fields are indicated in Figure 43-4. N(R) and N(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

   N(R) and N(S) values are presented in decimal format in modulo–8 traces. Each column displays a single digit that represents a 3–bit binary value. For modulo 128, the values $^{0}_{0}$ to $^{7}_{F}$ are given in "character" format, where the columns contain a two–digit hexadecimal character.

7. *P and F.* The status of the poll or the final bit is given in the **P/F** column. Whether this bit is the P– or F–bit is indicated for most frame types in Figure 43-4 (under "CONTROL").

   The setting of the P–bit in an INFO frame often denotes the retransmission of an unacknowledged frame following a T1 timeout.

8. *Size.* The number of bytes in each frame is given in this column in four decimal digits. The count begins with the first address byte and excludes the two–byte FCS. Frames without I–fields show a count of three (MOD 8).

9. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

- For character data

  Data To Record: CHARACTER
  Time Ticks: ON

  or

  Data To Record: CHARACTER
  Time Ticks: OFF
  Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
  Timestamp Resolution: 1mS or 1uS

  If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

10. *Frame Checking.* A LAPD frame ends as soon as a $^{7}_{E}$ flag or seven 1–bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

    A for abort is posted to the displays when a frame is ended by seven 1–bits.

43.3 Monitor Conditions

When the LAPD personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 43-5.

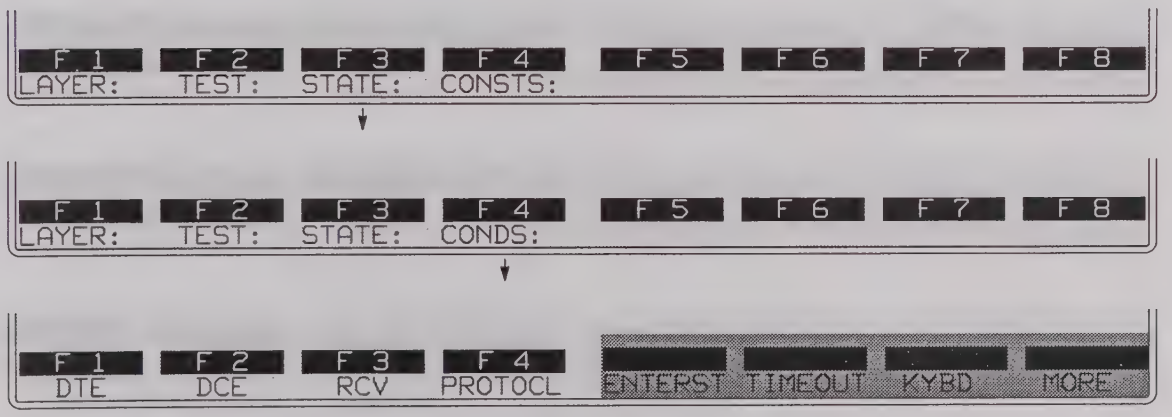


Figure 43-5 The softkeys for *DTE* and *DCE* are used to monitor LAPD protocol events at Layer 2.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of frames: INFO, SABM, UA, and so forth.

(A) Frame Types

The softkeys for INFO, supervisory, unnumbered, sequenced information, and "other" frames are illustrated in Figure 43-6. Press a softkey to write one of these frame types to the Layer 2 spreadsheet. DTE or DCE followed by a frame–type mnemonic—DTE INFO, for example, or DCE SABM—is a complete condition and will come true if a matching frame is monitored. SAPI, TEI, C/R, poll/final, and BCC conditions may be appended to the simple frame mnemonic, but they are optional.

1. *Info frames*. INFO frames differ for MOD 8 and MOD 128 numbering schemes. (See Figure 43-4.) For spreadsheet conditions to match I–frames accurately, the correct numbering system (**Mode of Operation**) should be selected on the LAPD Frame Level Setup screen.

2. *Supervisory frames*. The three supervisory–frame types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), and REJect. These frames always contain N(R) fields (see Figure 43-4) and serve mainly to acknowledge or reject INFO frames.

   Like INFO frames, supervisory frames are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

3. *Unnumbered frames*. Unnumbered frames generally assist in link–setup and takedown.

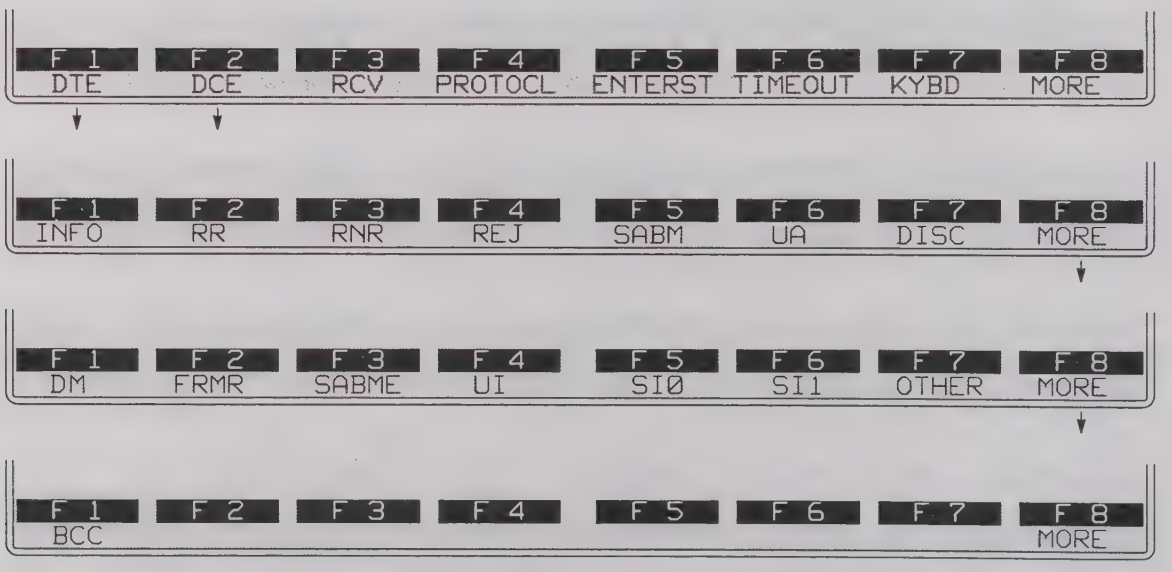


Figure 43-6 Frame type or block–check type may be specified as a complete condition for *DTE* or *DCE*.

4. *Sequenced information frames.* Sequenced information frames (SI0 and SI1) have a 1–bit sequence–numbering field that toggles 0 (SI0) and 1 (SI1). (See

Figure 43-4.) These frames are used instead of INFO frames in MOD 2 operation where the limit for outstanding information frames is 1.

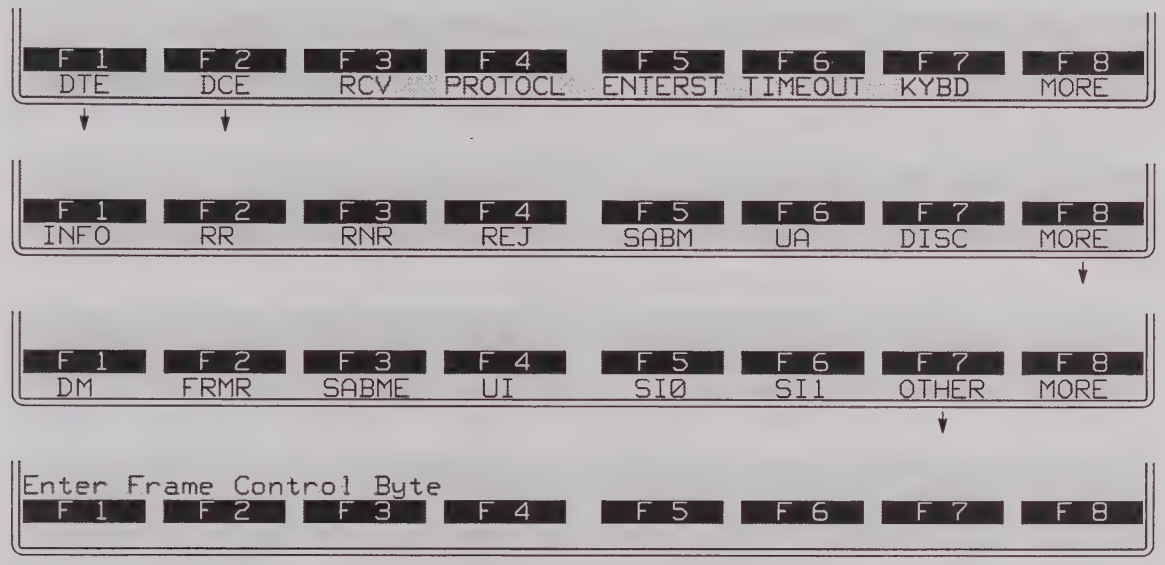


Figure 43-7 The hex value of any frame may be specified under *OTHER*.

5. *Other frames*. Any frame type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. See Figure 43-7. Then enter the hex byte in the form of two alphanumerics. Here, for example, is a SABM (with the P–bit set) entered as a hexadecimal:

CONDITIONS: DCE OTHER 3F

SAPI, TEI, C/R, P/F, and BCC conditions may be appended to OTHER conditions (see Figure 43-8). In MOD 8, the P/F bit is already specified in the hex entry, and a P/F condition will be ignored.

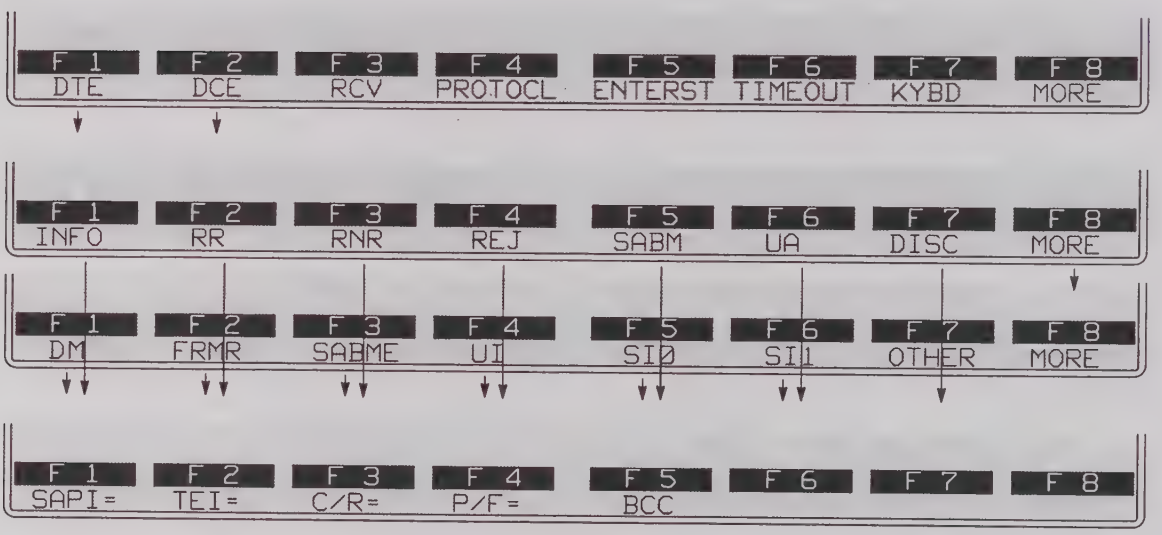


Figure 43-8 The bottom softkey rack shows conditions that may be linked to frame–type conditions.

6. *SAPI.* A SAPI condition may be added to all frame types. Press the softkey for SAPI=, shown in Figure 43-9. Then enter the 6–bit SAPI value as two hex digits in a range from 00 to 3F. (Do *not* use the [HEX] key.) The entry will appear as in this example:

CONDITIONS: DTE INFO SAPI= 15

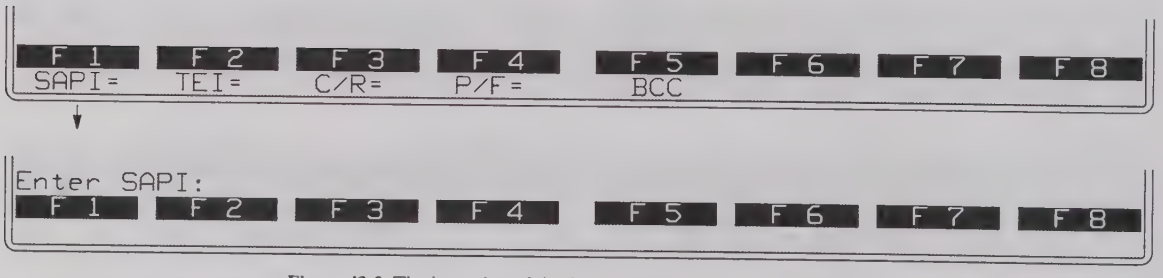


Figure 43-9 The hex value of the SAPI is entered as two alphanumerics.

To bypass the SAPI= selection (as well as the other options on the same rack of softkeys in Figure 43-9) press [DONE].

7. *TEI.* Like SAPI, a TEI condition may be added to all frame types. Press the softkey for TEI=, shown in Figure 43-10. Then enter the 7–bit TEI value as two hex digits in a range from 00 to 7F. (Do *not* use the [HEX] key.)

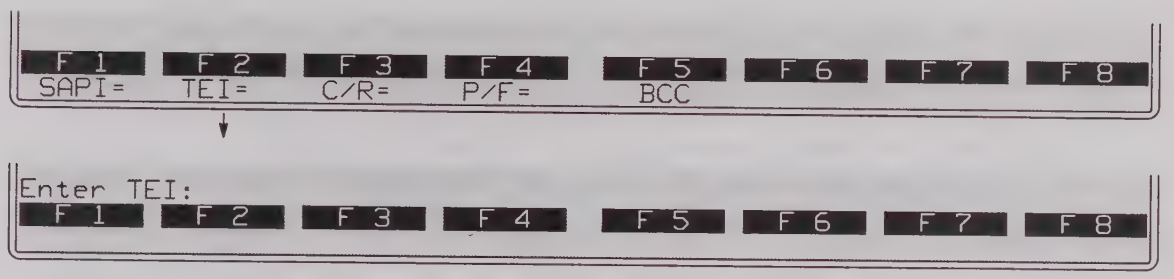


Figure 43-10 The TEI is an optional condition within all frame–type conditions.

8. *C/R.* A C/R value of 0 or 1 may be entered as an optional condition added to any frame–type condition.

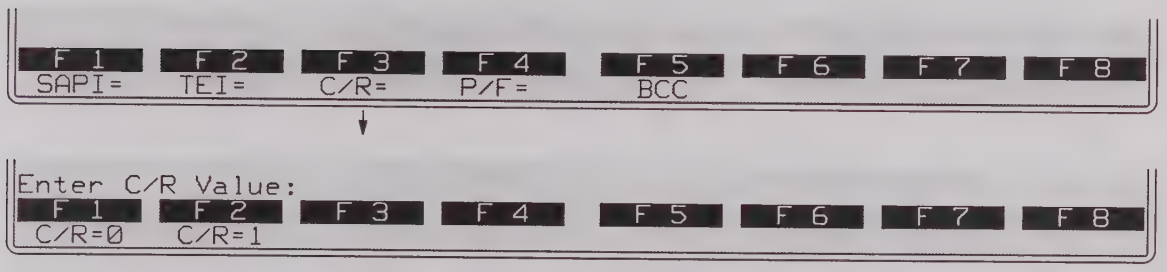


Figure 43-11 The value of the C/R bit may be chosen as a condition.

Press DONE to bypass the C/R= condition and the other conditions on the same softkey level in Figure 43-11.

9. *Poll/final bit.* P/F conditions are optional for all frame types. P/F values of 0 or 1 are entered by the softkey sequence in Figure 43-12.

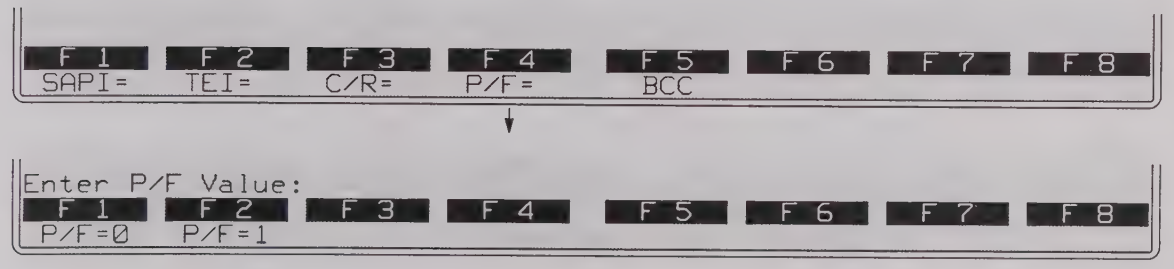


Figure 43-12 The value of the P/F bit may be chosen as a condition for any frame type.

Press DONE to bypass the P/F= condition and the other conditions on the same softkey level in Figure 43-12.

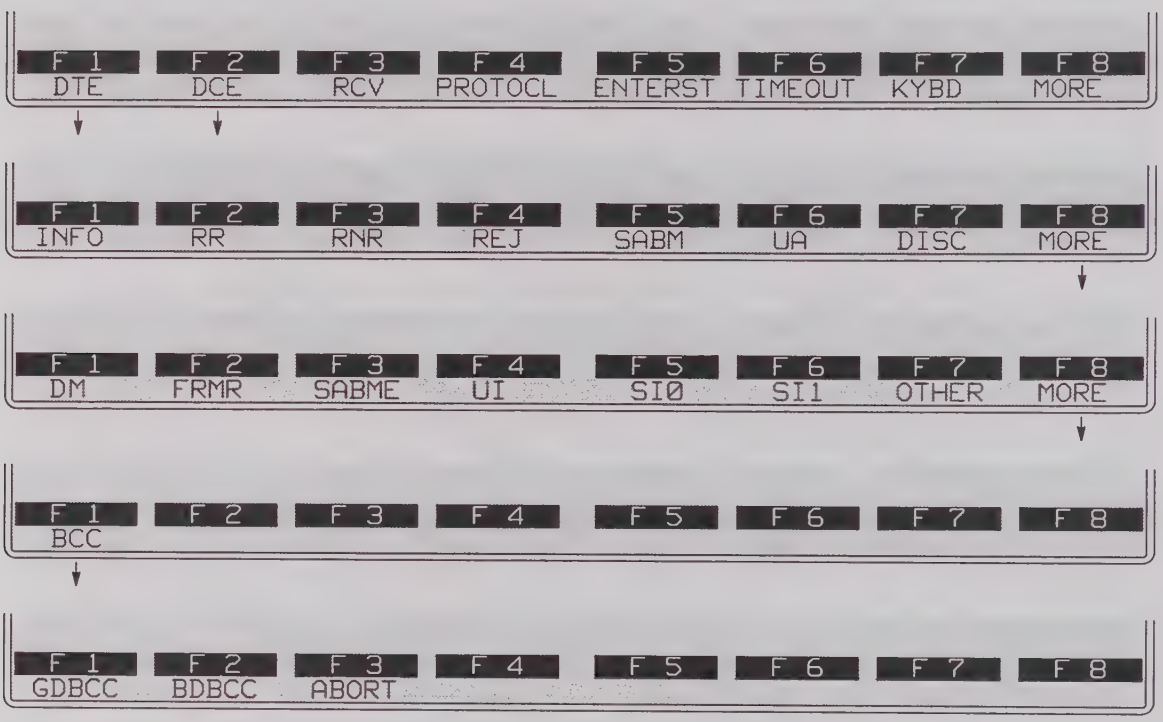


Figure 43-13 A condition may search for *all* good, bad, or aborted frames.

(B) BCC Conditions

DTE and DCE frames may be monitored for good and bad frame checks and for aborts. *All* DTE or DCE frames may be monitored with respect to frame checking, as in this example:

CONDITIONS: DTE BDBCC

The softkey sequence for this spreadsheet entry is given in Figure 43-13.

Or a particular *type* of frame may have a BCC or abort condition appended to it:

CONDITIONS: DCE INFO ABORT

43.4 Emulate–Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for **Mode:** EMULATE DTE or EMULATE DCE.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data lead for LAPD frame types. RCV conditions operate only in emulate modes, and they check only the data lead that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV INFO P/F=1 looks the same as DCE INFO P/F=1—there are important differences that are noted below.

1. *Valid frame sequencing.* To satisfy RCV conditions, numbered frames must have correct N(R) and N(S) sequencing.

2. *Good BCC.* RCV conditions cannot match frames with bad frame checks, nor can they match aborted frames. (Emulate–mode conditions are designed for ease of programming, and the assumption is that as a LAPD emulator, you are not required to acknowledge—or negative–acknowledge—bad or aborted frames.)

   If you wish to count bad BCCs or aborts, use DTE or DCE conditions instead of RCV conditions.

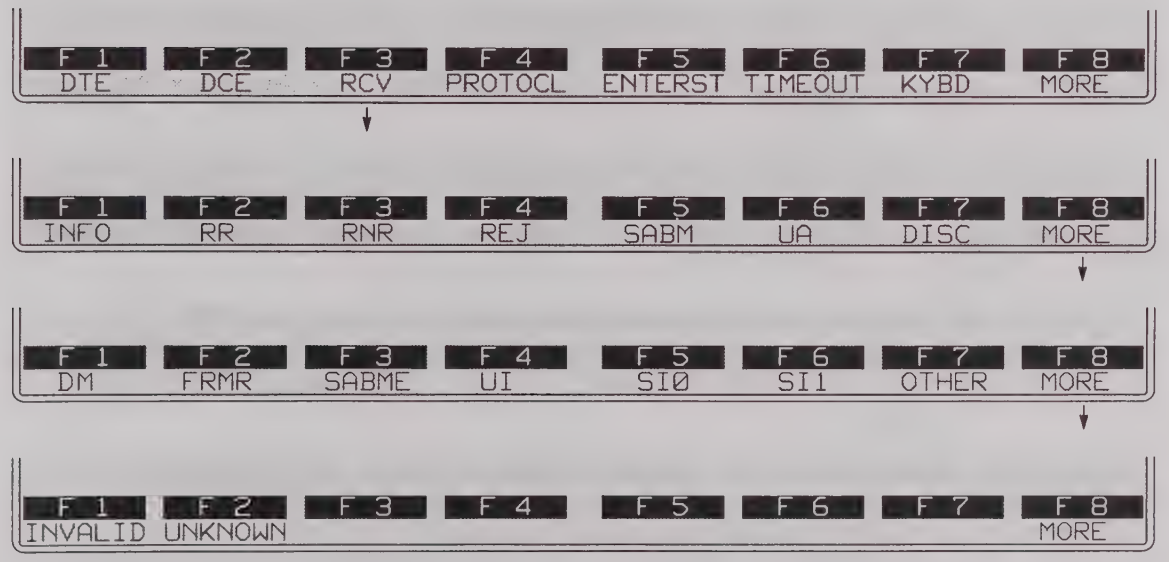


Figure 43-14 *INVALID* and *UNKNOWN* are frame types for *RCV* conditions.

3. *Type invalid.* RCV conditions can detect frames that are invalid "types"—the control field is missing, for example, or the I–field is missing in an I–frame. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 43-14.

4. *Type unknown.* A frame may be valid in all respects but have a control field that indicates a nonstandard frame type. Such a frame may be matched by a RCV UNKNOWN condition (Figure 43-14).

(B) N(S) Error

As a Layer 2 emulator, you do respond to INFO frames that have N(S) errors. These are detected as NS_ERR conditions, not as RCV INFO conditions.

NS_ERRs apply only to frames received when you are emulating. The same frame that triggers an NS_ERR condition also may satisfy a DTE INFO or DCE INFO condition—but not a RCV INFO condition.

NS_ERR will come true for any received INFO frame whose N(S) value is not one higher than the previous N(S).

NS_ERR will not come true for out–of–sequence SI0 and SI1 frames.

In the first rack of condition softkeys at Layer 2, press PROTOCL. Then press the softkey for NS_ERR. See Figure 43-15.

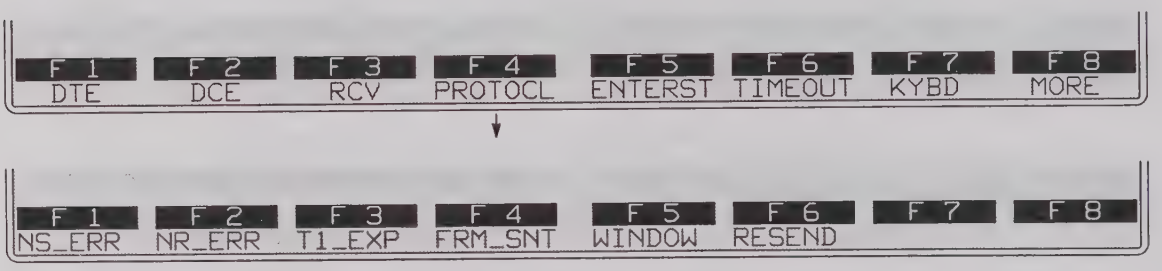


Figure 43-15 The *PROTOCL* key brings up six LAPD emulate conditions.

(C) N(R) Error

Received INFO or supervisory frames may have N(R) errors. Such errors are detected as NR_ERR conditions, not as RCV INFO or RR (or RNR or REJ) conditions.

A valid N(R) is any value that (1) acknowledges a frame that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other N(R) value is detected as an error.

(D) T1 Expired

This condition detects the expiration of the T1 timeout–timer that is regulated on the LAPD Frame Level Setup screen. See Section 43.1(A), above.

(E) Frame Sent

This condition is true when, as a result of a SEND or RESEND action, a frame has been passed down to Layer 1.

(F) Window Conditions

The size of the Layer 2 retransmit window is configured on the LAPD Frame Level Setup screen. See Section 43.1(D). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 43-16.

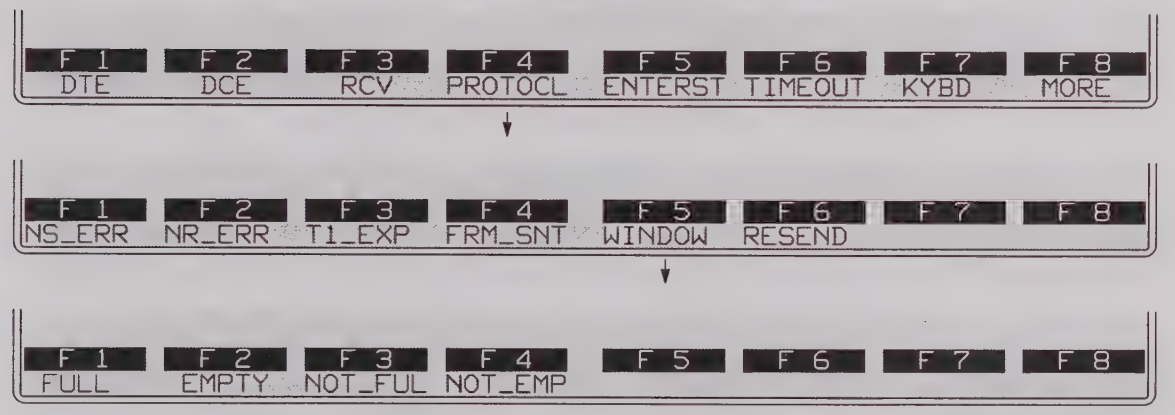


Figure 43-16 When the retransmit window fills, Layer 2 stops buffering frames for retransmission.

WINDOW FULL is true when the window is full of unacknowledged frames and the Layer 2 protocol package will not buffer additional frames until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no frames outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR
   WINDOW NOT_EMPTY
```

CAUTION: Window conditions are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as a RCV condition.*

(G) More to Resend

Frames in the window may have to be resent, usually as the result of a T1 timeout or a Reject frame. One RESEND action retransmits one frame in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent frames. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
    CONDITIONS: RCV REJ
    NEXT_ST: recover
STATE: recover
    CONDITIONS: ENTER_STATE
    ACTIONS: RESEND FIRST
    CONDITIONS: FRAME_SENT
        MORE_TO_RESEND
    ACTIONS: RESEND NEXT
    CONDITIONS: FRAME_SENT
        NO_MORE_TO_RESEND
    NEXT_ST: xfer
```

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 43-17.

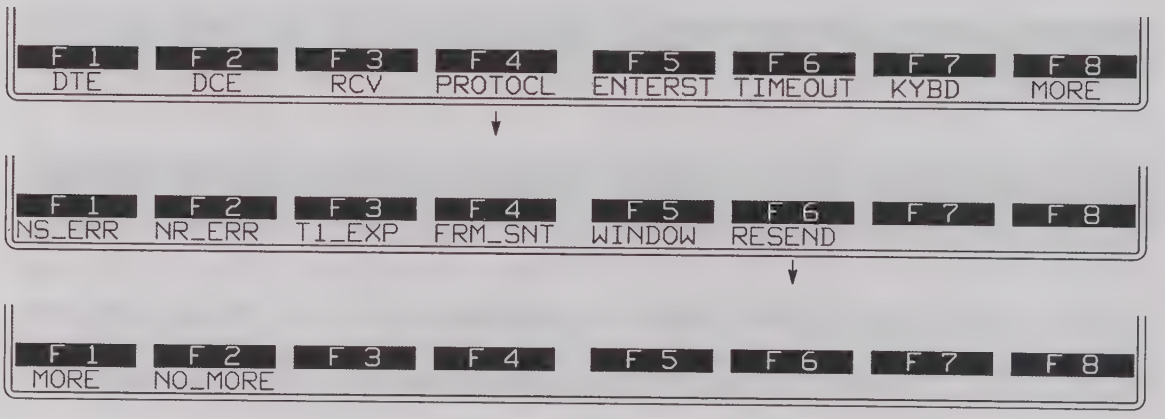


Figure 43-17 The *MORE_TO_RESEND* condition allows you to resend the entire window of frames and then stop when there are *NO_MORE_TO_RESEND*.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 31.2) and must *always be used in combination with a transitional condition such as FRAME_SENT.*

43.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 2, press DONE to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the LAPD personality package are shown in the racks of softkeys in Figure 43-18. Except for ENHANCE and SUPPRES, the actions shown have meaning only when the INTERVIEW is emulating DTE or DCE, and not when it is monitoring the line passively.

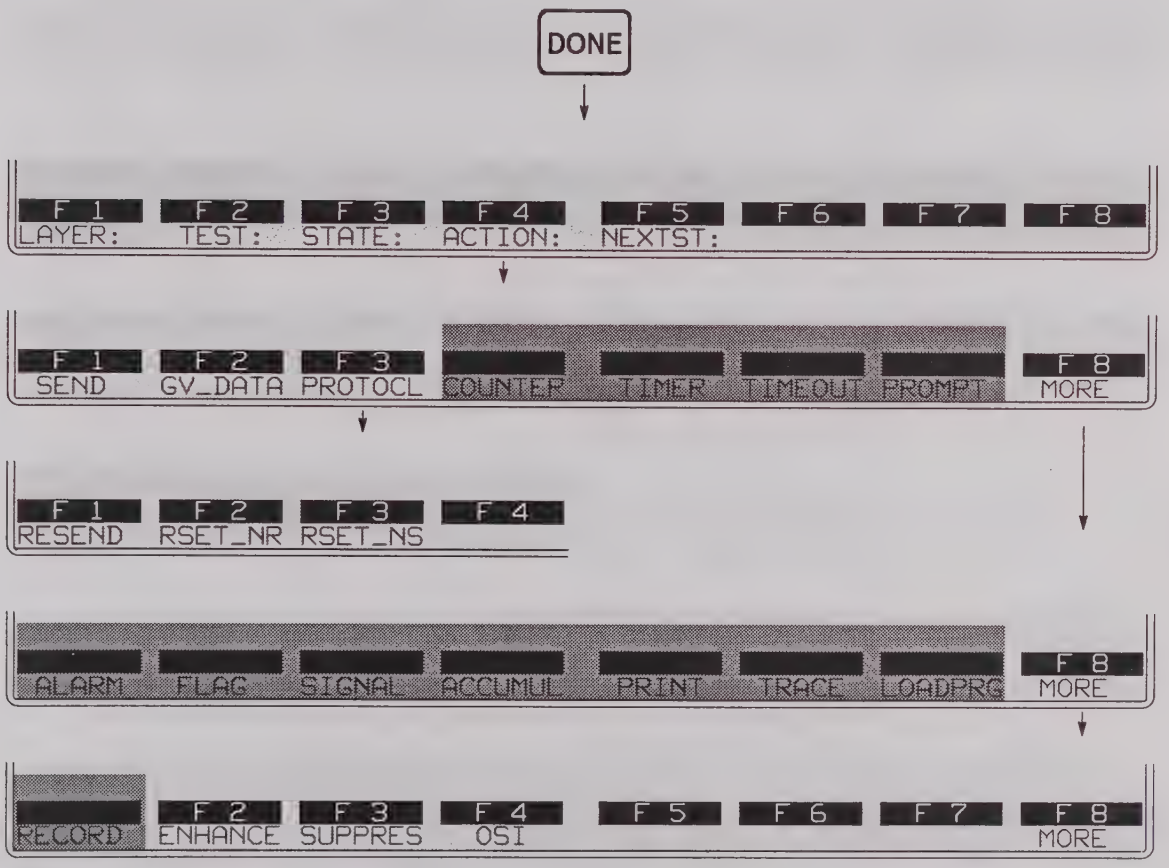


Figure 43-18 Action softkeys specific to LAPD.

(A) Send Actions

Press the softkey for SEND to access two racks of softkeys with names of frame types that may be named in SEND actions. All data generated by the LAPD package must be enclosed in a frame that is identified in a SEND action by *type*. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of frame types is given in Figure 43-19.

When conditions are true for a SEND action, frames are sent immediately down to Layer 1 to be transmitted there.

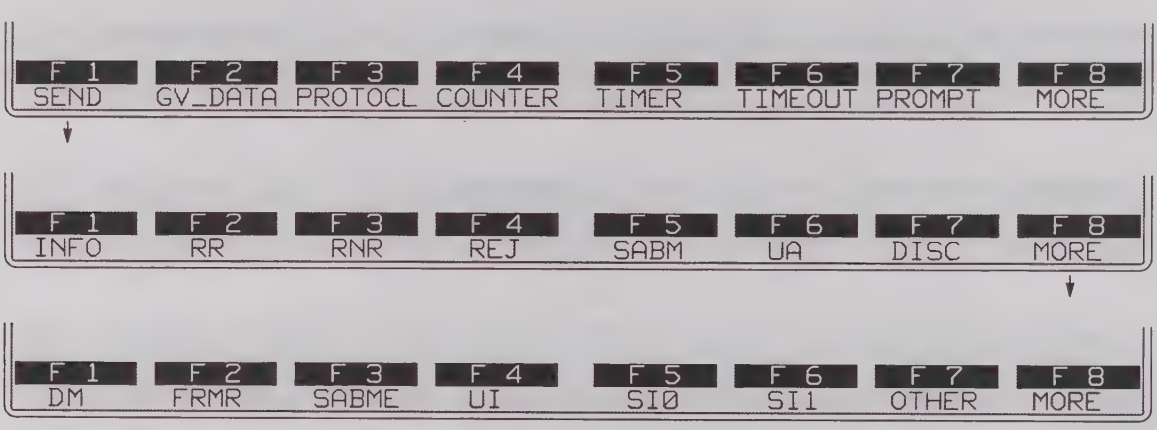


Figure 43-19 *SEND* actions always specify a frame type.

1. *INFO frames*. SEND INFO is a complete action–entry. SAPI, TEI, command–bit, poll–bit, N(R), N(S), string, and BCC parameters may be added to an INFO frame, but they are optional.

   If a Layer 3 package is installed and Layer 3 data is being handed down to Layer 2, the following condition–and–action trigger will accept this data and convey it properly to Layer 1:

   ```
   CONDITIONS: DL_DATA REQ
   ACTIONS: SEND INFO '' ((DL_DATA)) ''
   ```

   SEND INFO actions pass the INFO frame immediately to the next layer down. If the retransmit window is full, the frame is still sent—but it is not buffered in the window and cannot be *re*sent.

   An INFO frame will be buffered for retransmission regardless of the status of the window if a specific value is entered for the NS= parameter (see "*N(S)*," below). The specific N(S) value will clear the window and the INFO frame will be buffered in the first window position.

2. *Supervisory frames*. SEND RR, SEND RNR, and SEND REJ are complete action entries. SAPI, TEI, C/R bit, P/F bit, N(R), string, and BCC parameters may be added to the SEND action, but they are optional.

3. *Unnumbered frames*. SAPI, TEI, C/R bit, P/F bit, string, and BCC parameters values may be included as adjuncts to a SEND action for an unnumbered frame.

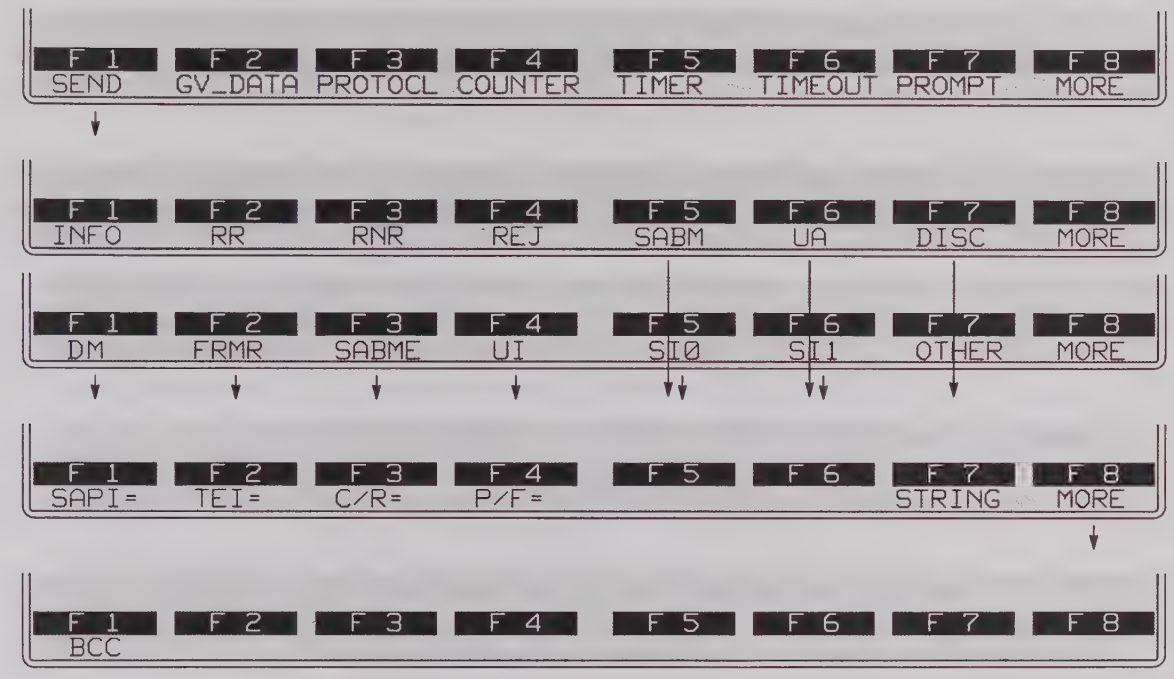


Figure 43-20 SAPI, TEI, C/R, P/F, string, and BCC options may be added to *SEND* unnumbered–type and SI–type actions.

4. *Sequenced information frames.* SAPI, TEI, C/R bit, P/F bit, string, and BCC values also may be added to SEND SI0 and SEND SI1 actions.

   Figure 43-20 shows the optional fields that may be specified inside unnumbered and SI send–actions.

5. *Other frames*. Any frame type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 43-19.

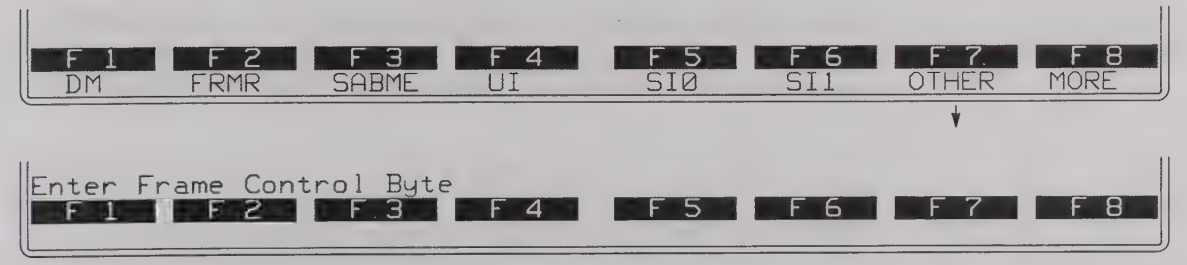


Figure 43-21 *SEND OTHER* actions always specify a *type* value in hex.

Enter the hex value in the form of two alphanumerics. Here is a DISConnect command entered as a SEND OTHER action:

SEND OTHER 43 SAPI= 04 TEI= 1A C/R= 0

Note that P/F, N(R), and N(S) fields are implied already in the user–entered hexadecimal control field. In MOD 128, P/F is not included in the hex entry and is a valid optional entry.

6. *SAPI.* An SAPI may be specified for all frame types. The SAPI= entry is always typed as two hex digits, with the right–hand digit expressing the four low–order bits of the SAPI and the left–hand digit expressing the two remaining bits. The SAPI field 000100, for example, appears as follows:

SEND RR SAPI= 04 TEI= 1A

7. *TEI.* A TEI may be specified for all frame types. The TEI= entry is always typed as two hex digits, with the right–hand digit expressing the four low–order bits of the TEI and the left–hand digit expressing the three remaining bits. The TEI field 0011010 is illustrated in the SEND RR example above.

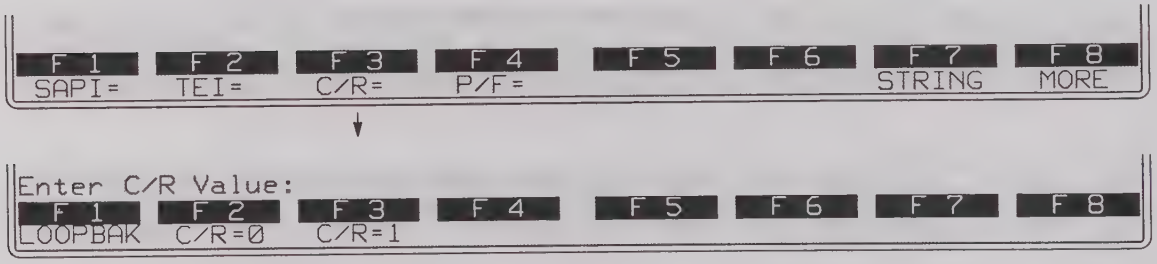


Figure 43-22 The correct Command/Response bit will depend on the logical emulation.

8. *Command/response bit.* The C/R bit may be specified for all frame types. The logical DTE uses C/R 0 for INFO frames and other command frames, and C/R 1 for responses. The logical DCE uses C/R 1 for INFO frames and other commands, and C/R 0 for responses. Since the response bit echoes the command bit, a C/R= LOOPBACK selection is provided (see Figure 43-22).

9. *Poll/final bit.* The P/F bit is an optional entry in all SEND actions. P/F values of 0, 1, or LOOPBAK are entered by the softkeys in Figure 43-23. If P/F= LOOPBAK, the bit will echo the last P/F bit received. (Looping the P/F bit is appropriate for UAs and supervisory frames.)

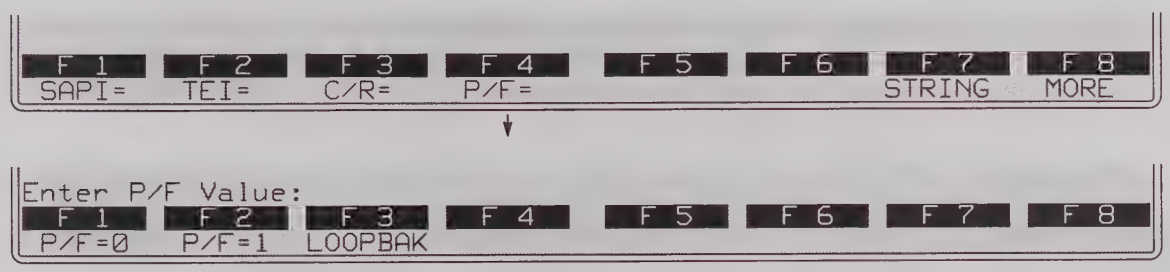


Figure 43-23 A P/F value is optional in all *SEND* entries.

10. *N(R)*. N(R) fields are transmitted in INFO and supervisory frames.

    To specify an N(R) value, press the softkey for NR= (see Figure 43-24). Enter a hexadecimal value written as one or two alphanumeric digits. For example, an entry that represented the highest valid N(R) in MOD 8 would be NR= 7. The highest valid entry in MOD 128 would be NR= 7F.

    Other N(R) options are ACK_NS, LAST_NR, and AUTO. (See Figure 43-24.) ACK_NS means that your N(R) will acknowledge (that is, it will be *one higher than*) the last N(S) value you received. Normally this will be the correct N(R), except in cases where the last N(S) received was erroneous. The NR= ACK_NS selection allows you to overlook N(S) errors.

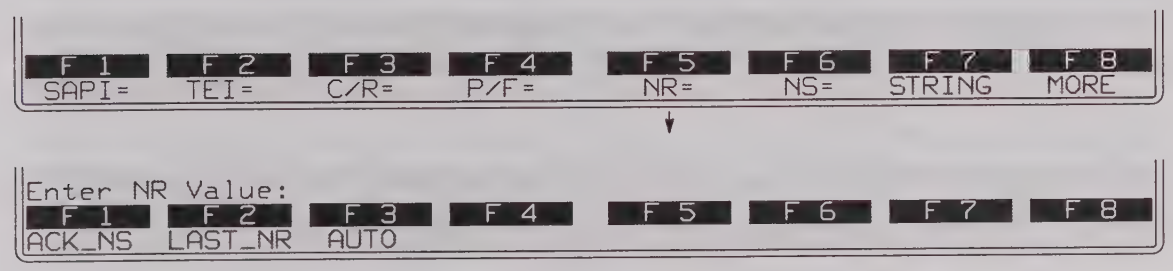


Figure 43-24 The N(R) field may be specified in INFO and supervisory frames to be sent.

LAST_NR means that you simply repeat the last N(R) you sent. Normally this is the correct N(R) following a bad N(S). The NR= LAST_NR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal LAPD station, ACKing valid N(S) values and repeating your last N(R) whenever an invalid N(S) is received.

11. *N(S)*. N(S) fields are transmitted in INFO frames only. (See the frame–field diagrams in Figure 43-4.) Entries for N(S) in SEND INFO actions are optional. The softkeys that open below NS= are illustrated in Figure 43-25.

To specify an N(S) value, press the softkey for NS=, then enter a hexadecimal in the form of one or two alphanumerics. Valid hex entries are the same as for N(R). A SEND INFO action that *specifies* an N(S) value—NS= 0, for example—will clear the window so that the INFO frame is buffered immediately.

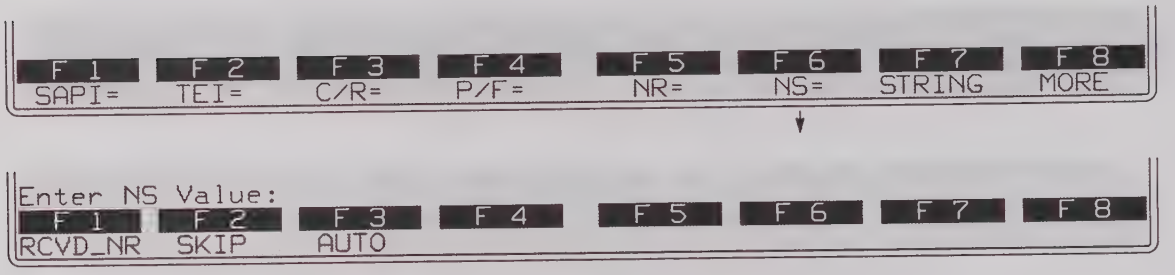


Figure 43-25 The N(S) field may be specified in a *SEND INFO* action.

Other N(S) options are RCVD_NR, SKIP, and AUTO. RCVD_NR means that you send the N(S) value that the other side says it is expecting. This is the valid N(S) in most cases, but not when you send two or more I–frames in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct N(S). This will look to the other side as though the line has taken a "hit" and a frame has been lost. This selection causes the window to be cleared.

NS= AUTO is the default setting for SEND INFO actions. AUTO means that every *new* INFO frame sent will have an N(S) value of one higher than the previous.

12. *String.* Strings are sent in LAPD only as adjuncts to frame–types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

    Press the SEND softkey followed by the softkey for a frame type. Add any necessary or desired SEND options for the particular frame type. Then press the STRING softkey (labeled F7 in Figure 43-25).

    There is no spreadsheet keyword that identifies send–strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " –character in your string, type \" .) See Section 33 for more information on strings.

    Here is a simple SEND action that includes no options besides a string:

    ACTIONS: SEND FRMR " $^{1}{}_{1}{}^{6}{}_{6}{}^{0}{}_{8}$ "

    And here is a SEND action that includes a full complement of optional fields, including a string:

```
ACTIONS: SEND INFO SAPI= 04 TEI= 1A C/R= 0 P/F= 0 NR= AUTO NS= AUTO
         "ᵀᴮ²ᶜ⁰ᴿ7036449000" GDBCC
```

Most ASCII−keyboard, control, and hexadecimal characters are legal in a send−string. Special keys ([FLAG], [DON'T CARE], [BIT MASK]) are not legal. Refer to Table 33-2.

To insert a canned fox message into a transmit string, type FOX inside double parens, as follows: ((FOX)) . Remember that the double parens are special characters produced by the [CTRL]−[9] and [CTRL]−[0] combinations. Constants, counters, and flags can also be embedded in a string. See Section 33, Strings.

13. *BCC.* There are three BCC options for every SEND action in LAPD. One of the options, GDBCC, is the default. Any frame that does not request a bad BCC or an abort will have a good frame−check sequence calculated for it and appended to it. BCC also is an option for SEND actions at Layer 1; but it does not occur at Layer 3 or higher.

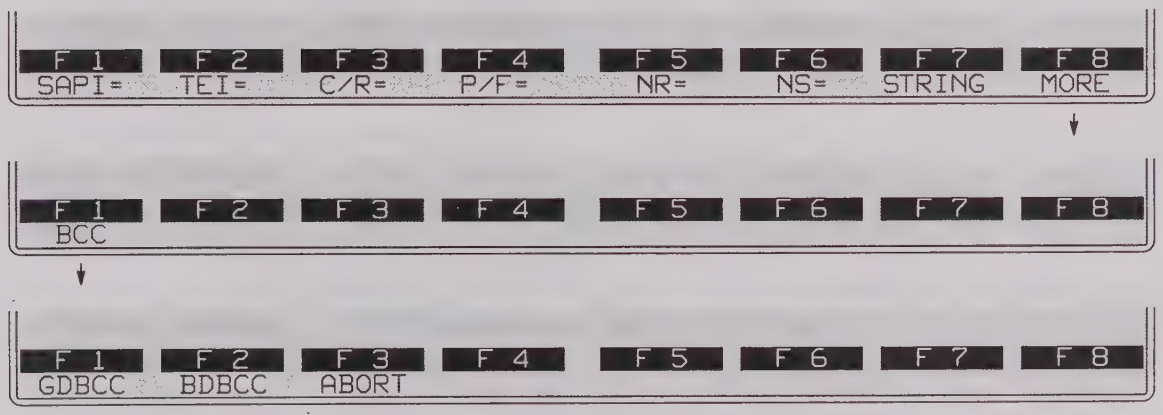


Figure 43-26 Type of BCC is a *SEND* option for frames at Layer 2.

The three softkey selections for BCC are shown in Figure 43-26. A 16−bit CCITT frame check is selected automatically for BOP protocols and cannot be changed or disabled. A bad BCC will be CRC−16 instead of CCITT.

When ABORT is the BCC selection, instead of appending a proper frame check the transmitter will hold the lead at mark for eight bits (or longer if the transmitter is idling^{F_F}). Inside a frame, seven 1−bits in a row are sufficient to signal an abort.

(B) Give Data

GIVE_DATA is the [F2] action on the first rack of action softkeys (refer to Figure 43-18). This action takes the I−field from a received INFO frame and passes it up to Layer 3 along with a DL_DATA IND primitive. (See Figure 34-5 in the section, OSI

Primitives on the Protocol Spreadsheet.) In an emulate mode, data is delivered up to Layer 3 only by one of two actions at Layer 2: GIVE_DATA, or else a DL_DATA IND primitive followed by the data string.

(C) Resend

The RESEND function is mapped to F1 on the second layer of action softkeys. See Figure 43-27. The first RESEND action will resend the first frame in the window. The window is a queue that buffers INFO frames for retransmission in case one or more transmissions are lost or in error.

The first frame in the window always is the *earliest outstanding* (unacknowledged) frame. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first–frame" position is established. The first RESEND after an acknowledgment always sends the first window frame.

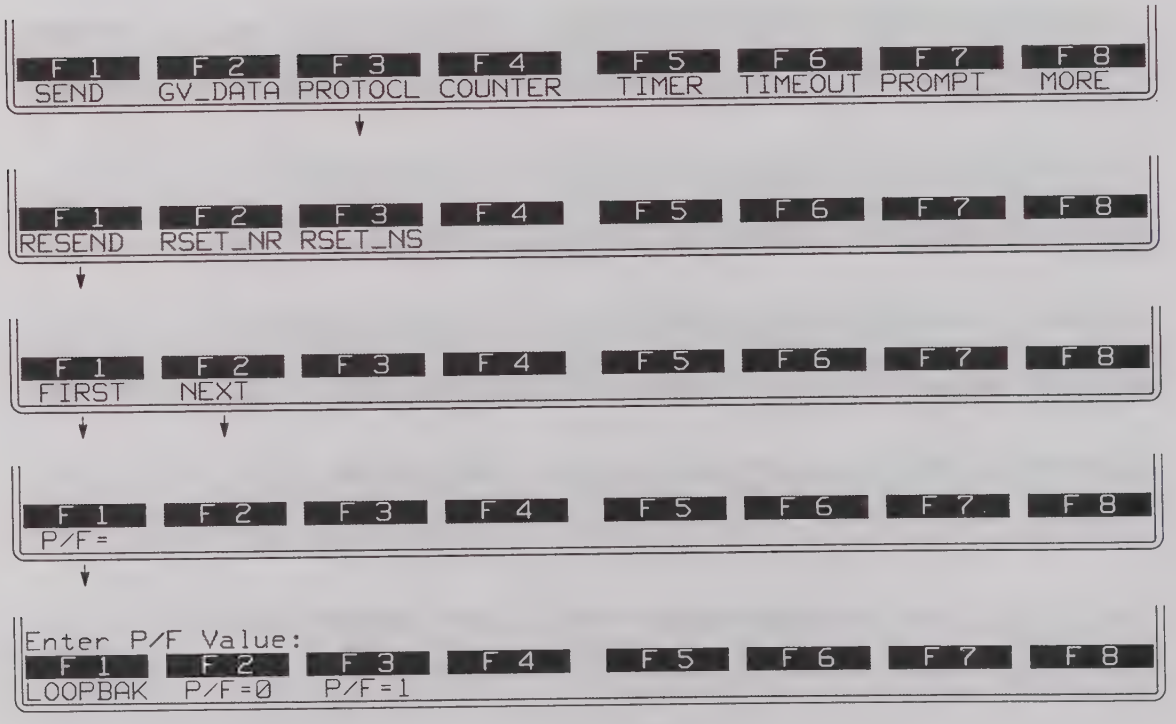


Figure 43-27 The *RESEND* action allows you to recover from sequence errors.

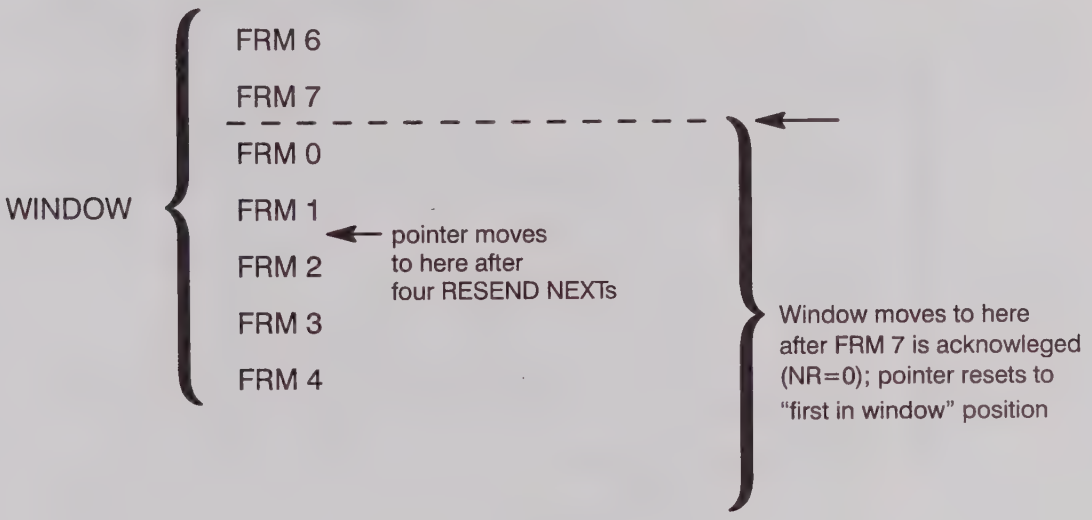


Figure 43-28 Resends cause the pointer to move, while acknowledgments move the pointer *and the entire window.*

The *second and subsequent* RESENDs following an acknowledgment also will send the first window frame, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window frames. Figure 43-28 shows the position of the the resend "pointer" after four consecutive RESEND NEXT actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

The resend–pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

1. *Resend first/next.* RESEND FIRST means that the resend–pointer is reset to the beginning of the window, the first frame in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 43-29.

   The RESEND FIRST action makes it possible for you to resend all the frames in the window one by one, and then *resend them again* if necessary.

2. *P/F=loopback/0/1.* The P/F bit in the resend–frame can be set to 0 or 1 by this optional action. If PF= LOOPBACK, the bit will echo the last P/F bit received. (Default is 1 in a RESEND action.)

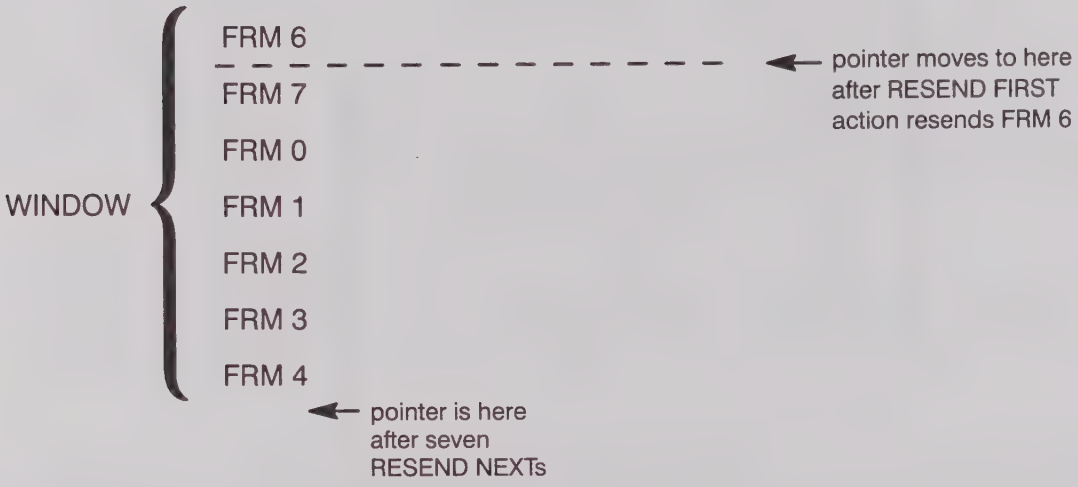


Figure 43-29 *RESEND FIRST* resets the pointer, allowing you to resend the entire window *repeatedly*.

(D) Reset N(R) and Reset N(S)

RESET_NR and RESET_NS are the F2 and F3 actions on the second rack of action softkeys in the LAPD personality package. (Refer again to Figure 43-18.) The sequence–number fields in I–frames and supervisory frames can be reset by these two Protocol Spreadsheet actions. Sequence numbers are not reset *automatically* during a test by any frame that is sent or received.

RESET_NS also clears the transmit window.

43.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 2 protocol trace (see Section 43.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the LAPD protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F2 on the third rack of action softkeys. Figure 43-30 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

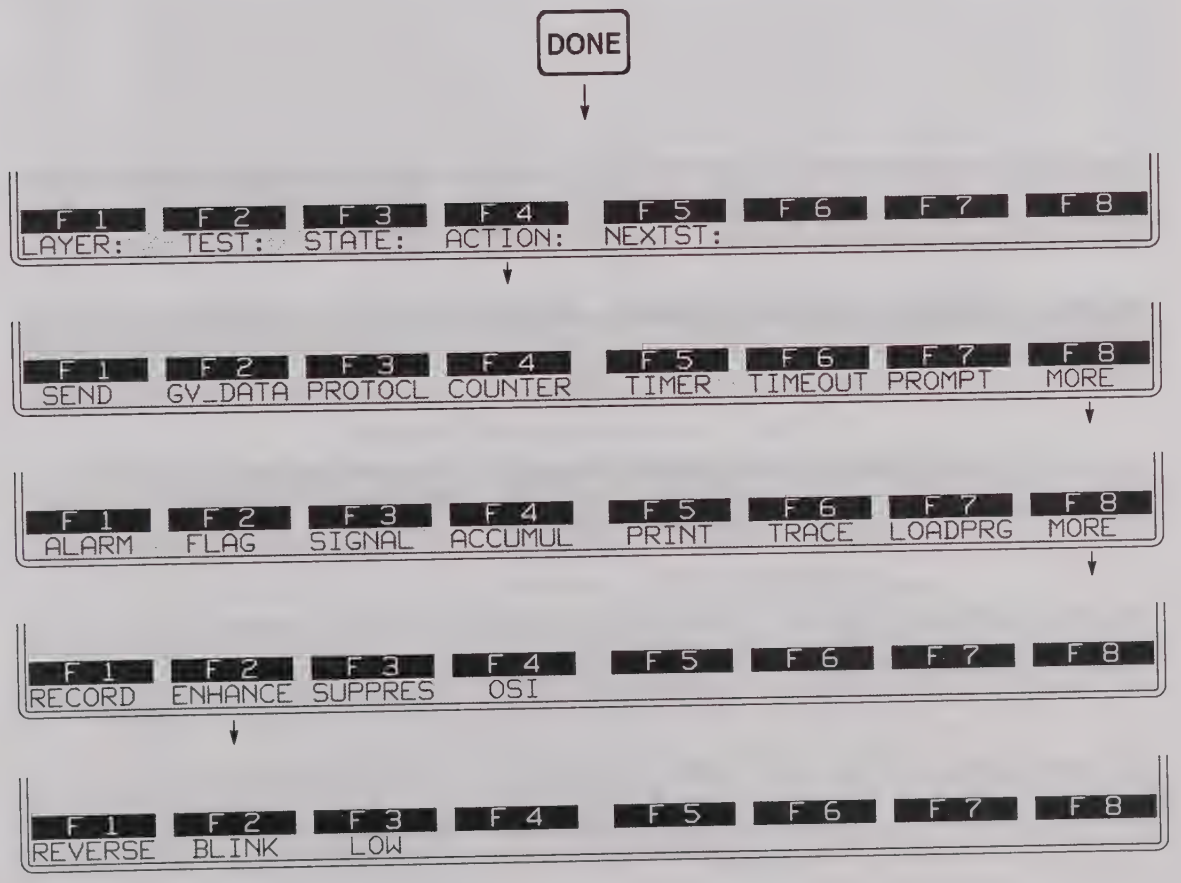


Figure 43-30 Selected frames on the protocol trace may be enhanced or suppressed.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

MON/DISK/FD1 BLK=00019 P 06/23/89 11:14
ASCII/8/NONE/BOP

SRC	SAPI	TEI	C/R	TYPE	Nr	Ns	P/F	SIZE	TIME	BCC
DCE	04	1A	1	DISC			0	0003	0915:08.021	G
DTE	04	1A	1	UA			0	0003	0915:08.172	G
DTE	04	1A	0	SABM			1	0003	0915:08.212	G
DCE	04	1A	0	UA			1	0003	0915:08.440	G
DTE	04	1A	0	INFO	0	0	0	0007	0915:08.482	G
DCE	04	1A	1	INFO	0	0	0	0008	0915:08.723	G
DTE	04	1A	1	RR	1		0	0003	0915:08.842	G
DCE	04	1A	0	RR	1		0	0003	0915:08.915	G
DTE	04	1A	0	INFO	1	1	0	0019	0915:08.916	G
DCE	04	1A	0	RR	2		0	0003	0915:08.959	G
DCE	04	1A	1	INFO	2	1	0	0006	0915:09.033	G
DTE	04	1A	1	RR	2		0	0003	0915:09.362	G
DCE	04	1A	1	INFO	2	2	0	0018	0915:09.505	G
DTE	04	1A	1	RR	3		0	0003	0915:09.667	G
DTE	04	1A	0	INFO	3	2	0	0006	0915:09.747	G

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8
L2TRACE L3TRACE

Figure 43-31 A DTE SABM has been enhanced.

Figure 43-31 shows one screen of a Layer 2 protocol trace in which DTE SABM frames have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE SABM
ACTIONS: ENHANCE REVERSE

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 43-30 showed the softkey path to SUPPRES.

43.7 Automatic Primitives

In Section 34,OSI Primitives on the Protocol Spreadsheet, Table 34-2 lists the OSI service primitives that are monitored at the boundaries of Layer 2 as trigger conditions and sent up to Layer 3, or down to Layer 1, as user–entered spreadsheet actions. These primitives are layer–specific rather than protocol–specific and are not part of the personality package for LAPD; but a few of the primitives are set in motion automatically by LAPD spreadsheet actions. These *automatic* primitives can be thought of as part of the Layer 2 actions themselves, and by extension, as part of the LAPD protocol package.

Table 43-1 gives the set of LAPD actions that have action–primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 2, a DL_DATA IND primitive is forwarded to Layer 3, where a DL_DATA IND condition may be waiting to monitor it.

Whenever a SEND or RESEND action is initiated at Layer 2, a PH_DATA REQ primitive is sent downward along with the PH data (the entire frame).

If a SEND or RESEND action is triggered at Layer 2 while the physical connection at Layer 1 is inactive, Layer 2 will sense the absence of a physical connection and delay the PH_DATA REQ. Instead, it will send a PH_ACTIVATE REQ primitive. Only when a PH_ACTIVATE CONF has been returned by Layer 1, will Layer 2 release the data and the data primitive.

> **NOTE:** All Layer 1 interfaces for the INTERVIEW 7000 Series will return PH_ACTIVATE CONF automatically whenever they see PH_ACTIVATE REQ.

Table 43-1
Automatic Primitives Generated in LAPD

LAPD Layer 2 Action	Automatic Primitive	To Layer
GIVE_DATA	DL_DATA IND	3
SEND {TYPE}	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1
RESEND	(PH_ACTIVATE REQ*)	1
	PH_DATA REQ	1

*Sent if Layer 1 shows inactive status. PH_DATA REQ delayed until PH_ACTIVATE CONF returned by Layer 1.

44 Q.931

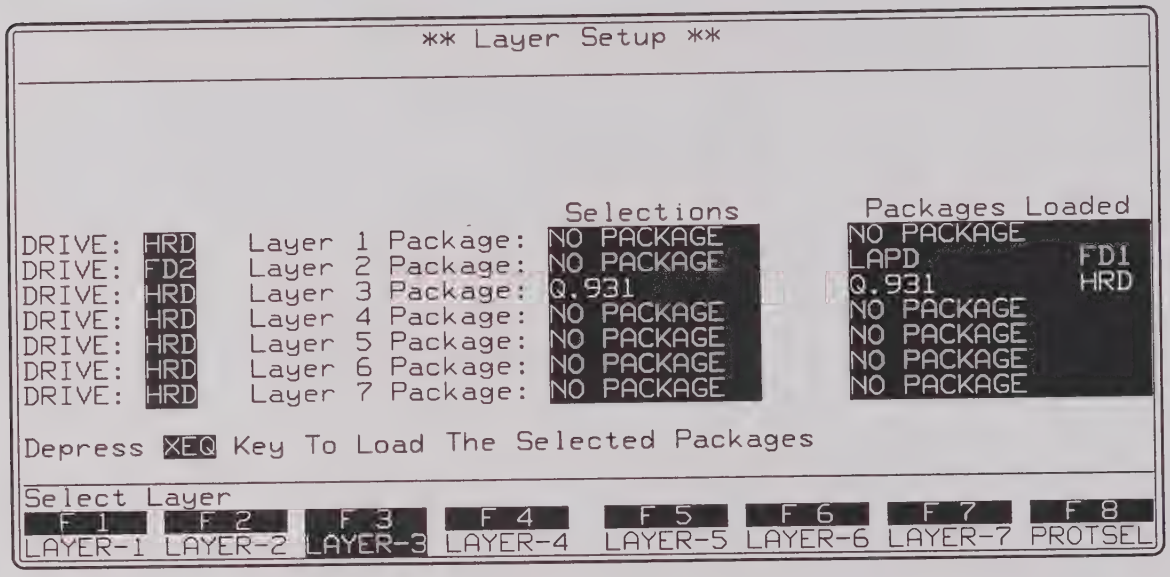


Figure 44-1 The Q.931 personality package is loaded from the Layer Setup screen at Layer 3.

44 Q.931

Q.931 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 44-1 shows the Layer Setup screen configured to load in the Q.931 package from the hard disk. Refer to Section 8 for details on operating the Layer Setup screen.

The Q.931 package consists of the following:

- A protocol trace (illustrated in Figure 44-2) that distills, from Q.931 data, the Level 3 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.

- A group of conditions and actions at Layer 3 on the Protocol Spreadsheet that facilitate ISDN programming. Figure 44-4 shows the softkey path to the first rack of condition softkeys when the Q.931 package is loaded in at Layer 3 and a LAYER: 3 programming block has been opened on the spreadsheet.

44.1 Protocol Trace

The Q.931 package includes an automatic message–trace display that summarizes Layer 3 activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

While the unit is in Run mode, press the softkey for PROTOCL ([F2] on the primary rack of display–mode softkeys) and then the softkey for L3TRACE ([F2]) to bring the protocol trace for Q.931 to the screen. Figure 44-2 is an example of this trace display. Each horizontal row in the trace represents a "message" as defined in CCITT Recommendation Q.931.

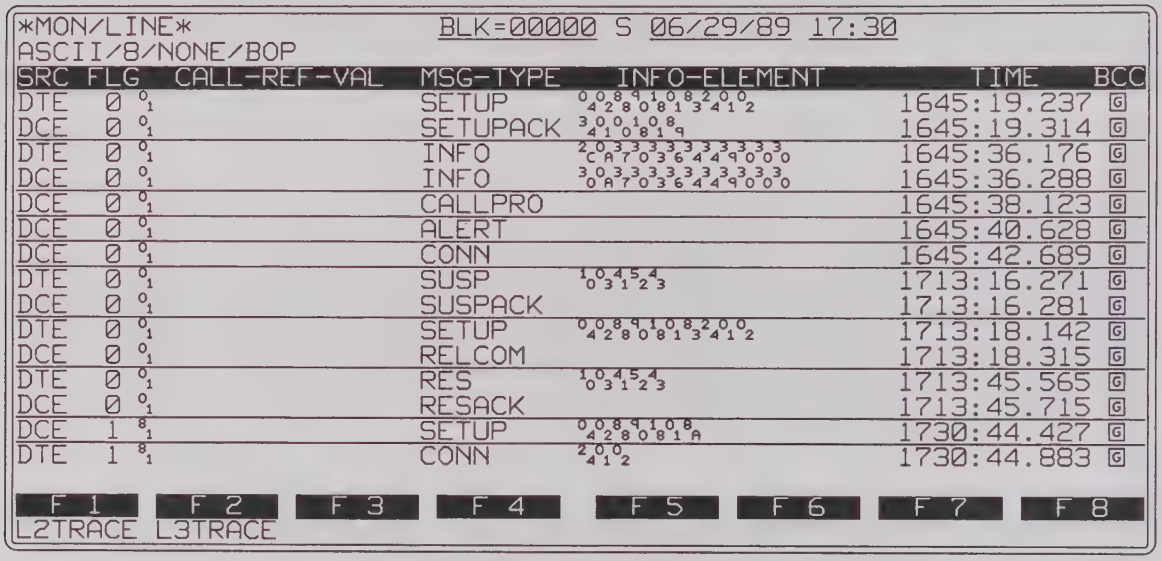


Figure 44-2 Each horizontal row on the trace display represents a message.

(A) The Protocol Trace in Freeze Mode

Press FREEZE to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing ROLL FWD or ↓ moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing ROLL BACK or ↑ moves the viewing window up to add a line of older data to the top of the screen.

Depression of the NEXT PAGE key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of PREV PAGE adds 15 lines of older frames.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys.

(B) Trace Columns

The columns in the protocol trace for Q.931 are explained below.

1. *Source*. The **SRC** column identifies the lead on which the message was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the message, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

   Just as on the data display, RD data is underlined.

2. *Flag*. The **FLG** column refers to the call–reference flag and is used to identify where the call originated. Calls originating on the user side of the link will show a flag value of 0 in every message from both the DTE and the DCE. (This is a *call*–origination flag, not a *message*–origination flag.)

   If a call originates from a remote user, every message that references that call will have a flag value of 1.

3. *Call reference value*. The number used by a message to reference a particular call is given in the **CALL–REF–VAL** column. This is a 15–column field that displays the variable–length call–reference value in from one to 15 hexadecimal characters. This field is blank when the call–reference *length* value (see Figure 44-3) is zero.

   The flag bit (see "Flag," above) is not included in the display of the first byte of the call–reference value.

4. *Message type*. The thirty message types that are named in the **MSG–TYPE** column are shown in Figure 44-3. If a message type does not fit any of the patterns in the figure, it is listed in the **MSG–TYPE** column as UNKWN followed by the hexadecimal value of the message–type byte: UNKWN=7C.

   If the number of bytes in the message is below the required minimum, the message is posted as INVALID.

5. *Information element*. The **INFO–ELEMENT** field presents up to 16 bytes of character data (decoded into hex) for all messages that contain information elements beyond the message–type field. Info elements are mandatory in some messages and optional in others.

   In the **INFO–ELEMENT** fields with data shown in Figure 44-2, the first byte always is an "identifier" byte, with a value that you can decode using the "Info Element 1" field diagrams on the right half of Figure 44-3. The second byte is a length byte, with a value that indicates the number of "contents" bytes that intervene before the next element–identifier.

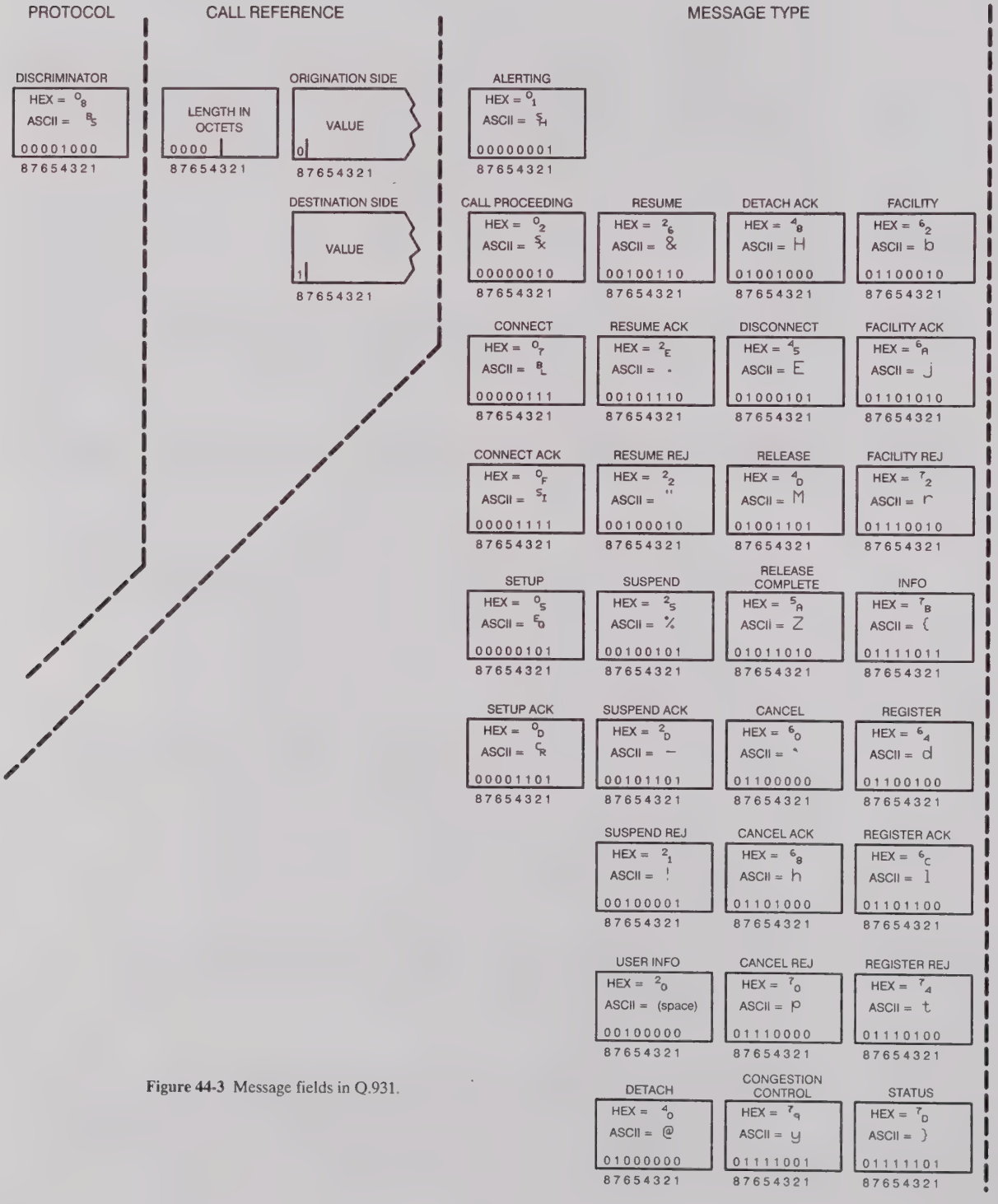
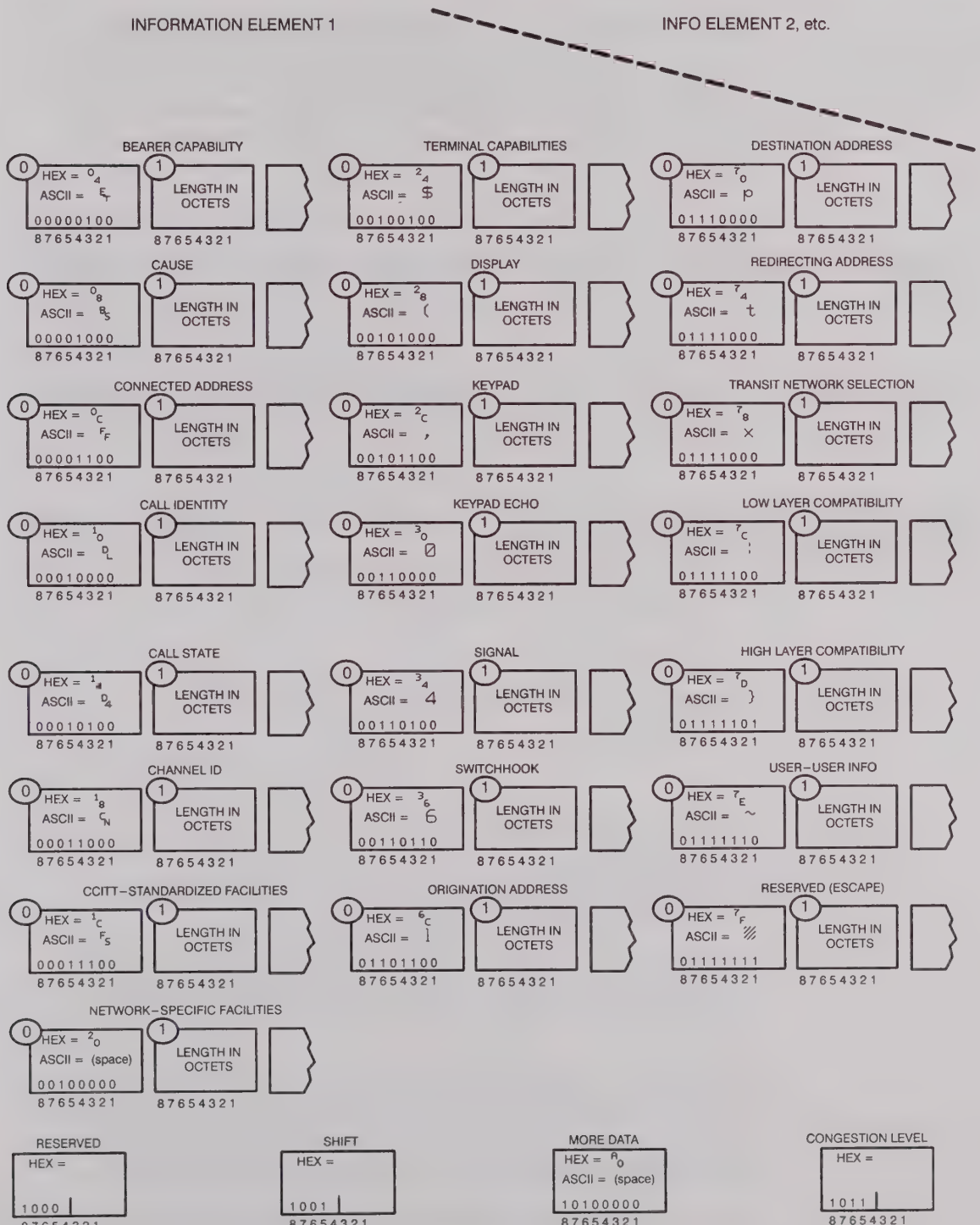


Figure 44-3 Message fields in Q.931.


INFORMATION ELEMENT 1
INFO ELEMENT 2, etc.
BEARER CAPABILITY
HEX = 04
ASCII = ET
00000100
LENGTH IN OCTETS
TERMINAL CAPABILITIES
HEX = 24
ASCII = $
00100100
DESTINATION ADDRESS
HEX = 70
ASCII = p
01110000
CAUSE
HEX = 08
ASCII = BS
00001000
DISPLAY
HEX = 28
ASCII = (
00101000
REDIRECTING ADDRESS
HEX = 74
ASCII = t
01111000
CONNECTED ADDRESS
HEX = 0C
ASCII = FF
00001100
KEYPAD
HEX = 2C
ASCII = ,
00101100
TRANSIT NETWORK SELECTION
HEX = 78
ASCII = X
01111000
CALL IDENTITY
HEX = 10
ASCII = DL
00010000
KEYPAD ECHO
HEX = 30
ASCII = 0
00110000
LOW LAYER COMPATIBILITY
HEX = 7C
ASCII = |
01111100
CALL STATE
HEX = 14
ASCII = D4
00010100
SIGNAL
HEX = 34
ASCII = 4
00110100
HIGH LAYER COMPATIBILITY
HEX = 7D
ASCII = }
01111101
CHANNEL ID
HEX = 18
ASCII = CN
00011000
SWITCHHOOK
HEX = 36
ASCII = 6
00110110
USER–USER INFO
HEX = 7E
ASCII = ~
01111110
CCITT–STANDARDIZED FACILITIES
HEX = 1C
ASCII = FS
00011100
ORIGINATION ADDRESS
HEX = 6C
ASCII = l
01101100
RESERVED (ESCAPE)
HEX = 7F
ASCII = ///
01111111
NETWORK–SPECIFIC FACILITIES
HEX = 20
ASCII = (space)
00100000
0
1
87654321
RESERVED
HEX =
1000
SHIFT
HEX =
1001
MORE DATA
HEX = A0
ASCII = (space)
10100000
CONGESTION LEVEL
HEX =
1011

The first info–element data in Figure 44-2, for example, begins with the hex character °₄. This translates on the field diagrams in Figure 44-3 as a "Bearer Capability" info element. The next byte, °₂, indicates that the remaining contents are two bytes long, followed by the end of the data or by another information element. This means that ¹₈, "Channel ID," is the next info–element identifier in the string; and so forth.

Note in the field diagrams in Figure 44-3 that there are four types of single–byte information element.

6. *Time.* The time of the arrival of the *end of the frame containing the message* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

   If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

7. *Frame Checking.* A BOP frame ends as soon as a ⁷ₑ flag or seven 1–bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol [G] denotes a good frame check, while [B] symbolizes a bad frame.

   [A] for abort is posted to the displays when a frame is ended by seven 1–bits.

44.2 Monitor Conditions

When the Q.931 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 3. See Figure 44-4.

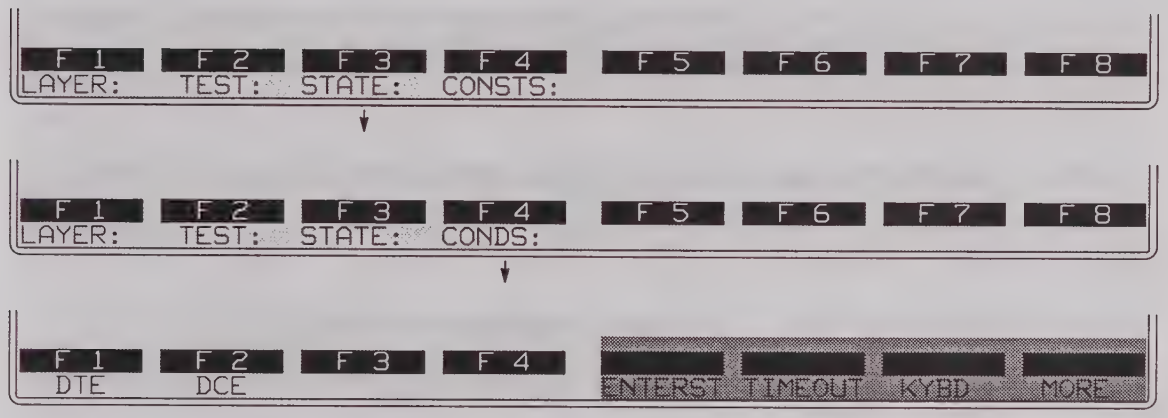


Figure 44-4 The softkeys for *DTE* and *DCE* are used to monitor ISDN protocol events once *Layer: 3* has been entered on the Protocol Spreadsheet.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of message: ALERT, CONN, STATUS, and so forth.

(A) Message Types

The softkeys for the thirty standard as well as "other" message types are illustrated in Figure 44-5. Press a softkey to write one of these message types to the Layer 3 spreadsheet. DTE or DCE followed by a message–type mnemonic—DTE SUSP, for example, or DCE REL—is a complete condition and will come true if a matching message–type is monitored. Call–reference, origination–link, and destination–link conditions may be added to the simple message–type mnemonic, but they are optional.

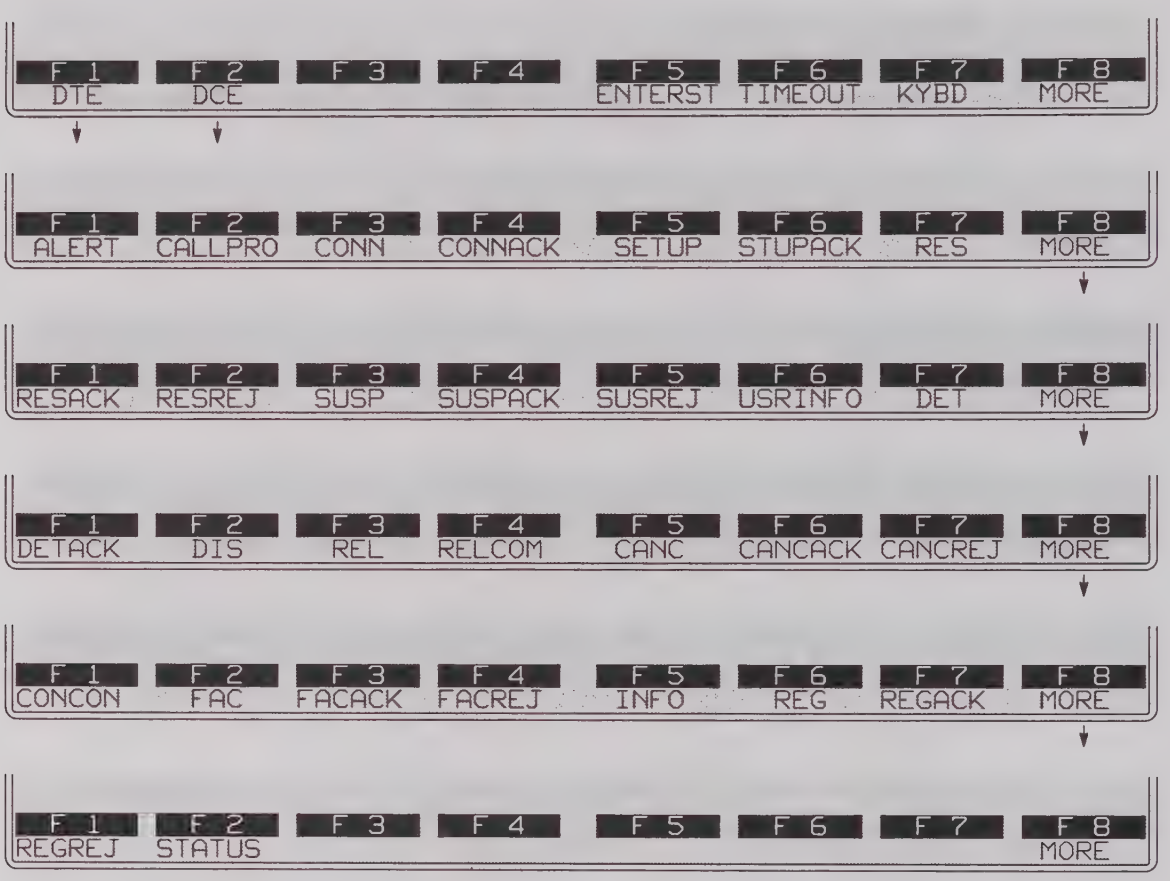


Figure 44-5 Message types.

(B) Call Reference Value

A specific call–reference value may be added as a condition to any of the message type conditions. Once you have pressed the softkey for a particular message type (or once you have touch–typed the message type followed by a space), the rack of softkeys shown in Figure 44-6 will appear.

Press the softkey for C_REF=, shown in Figure 44-6. Then enter the call–reference value as a sequence of hex digits inside quotation marks. The sequence may be from one to twelve semioctets long. Type each digit as an alphanumeric in the range *0–9* and *A–F* (or *a–f*), *without using the* [HEX] *key*. The call reference $^{7}{}_{F}$ with flag bit 0, for example, appears as follows:

CONDITIONS: DTE SETUP C_REF= "7F"

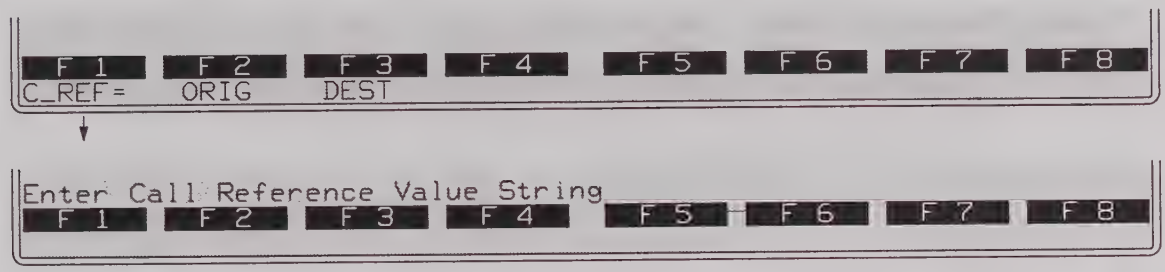


Figure 44-6 Call–reference, origination–link, and destination–link conditions may be added to a message–type condition.

Include the origination/destination–*flag* bit in your string. If the DCE, for example, gives call reference #4 to a call that originated at a remote link, the entry C_REF= "84" (with flag = 1) will detect this call reference, while the string "04" (flag = 0) will not detect it.

(C) Origination/Destination Link

A message–type condition may be set to come true only with respect to calls that originated *locally* (or *remotely*).

To make a message–type condition specific to calls that originated with a user on the link that is being monitored by the INTERVIEW, press the softkey for ORIG (F2 on the rack of softkeys in Figure 44-6). Only messages that have zero as the call–reference flag bit will satisfy this condition.

This condition, for example, looks for a Facility–type message that references a call that originated locally:

CONDITIONS: DTE FAC ORIG

Or a message–type condition may require remote origination of a call. In that case, the link being monitored is the "destination" of the call, and you will press the softkey for DEST, F3 on the rack of softkeys in Figure 44-6. Only messages that have 1 as the call–reference flag bit will satisfy this condition.

44.3 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 3 protocol trace (see Section 44.1). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE and DCE conditions can trigger an ENHANCE or SUPPRESS action.

(A) Enhance

Whenever a DTE or DCE condition comes true at Layer 3, the message that satisfied the condition can be enhanced on the Q.931 protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F1 on the first rack of action softkeys. Figure 44-7 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

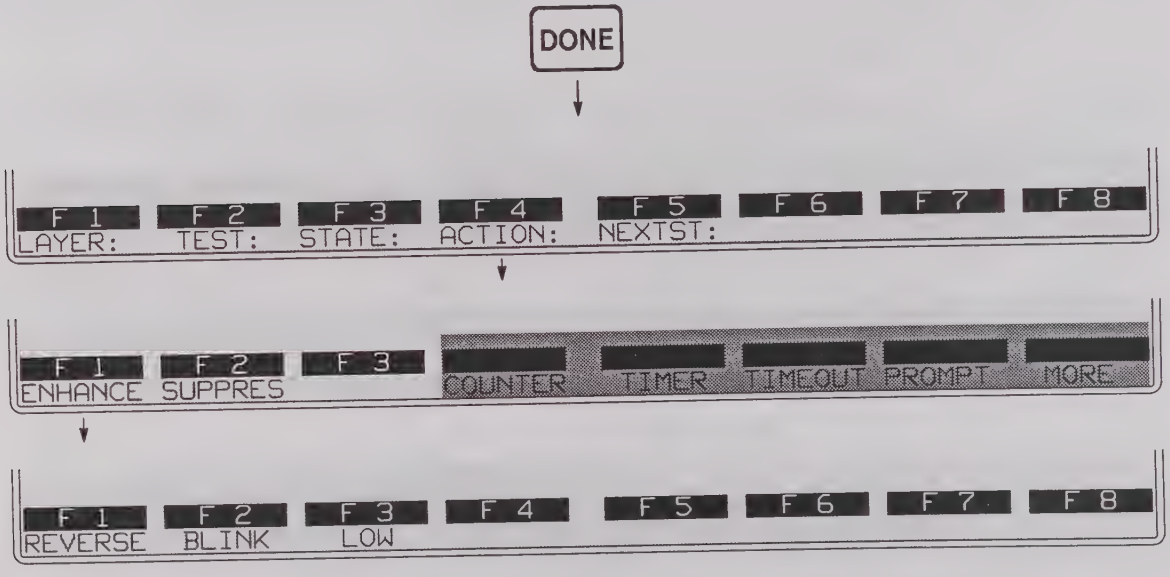


Figure 44-7 Selected messages on the protocol trace may be enhanced or suppressed.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

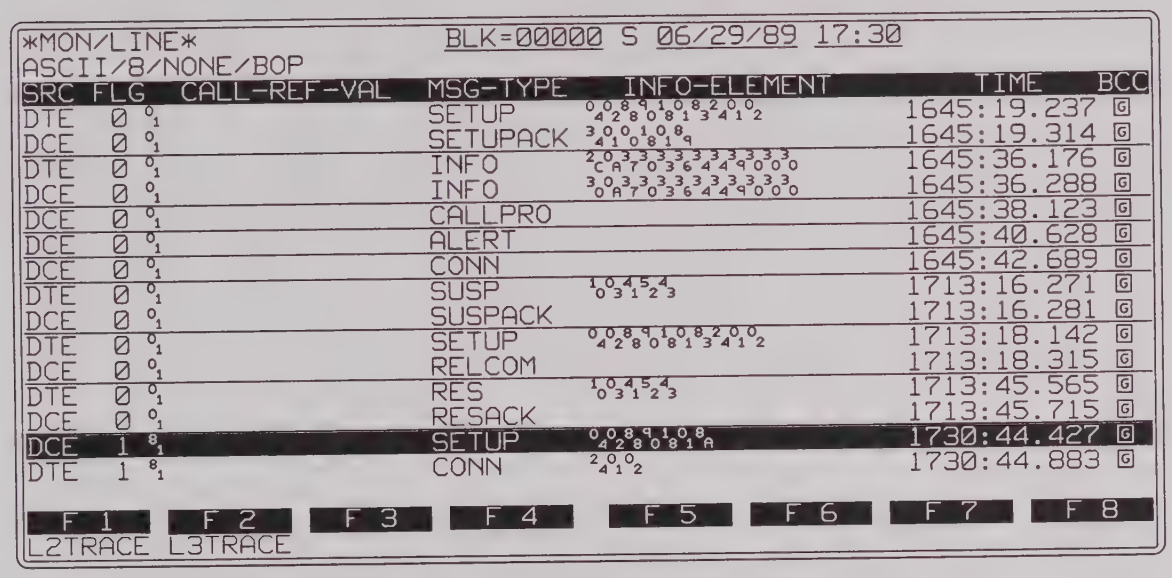


Figure 44-8 A DCE SETUP has been enhanced.

Figure 44-8 shows one screen of a Layer 3 protocol trace in which DCE SETUP messages have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

```
CONDITIONS: DCE SETUP
ACTIONS: ENHANCE REVERSE
```

(B) Suppress

Individual message–types that are suppressed in Layer 3 actions are deleted from the trace display. Here is an example of an action that suppresses all DTE Information messages:

```
CONDITIONS: DTE INFO
ACTIONS: SUPPRESS
```

45 SS#7 Layer 1

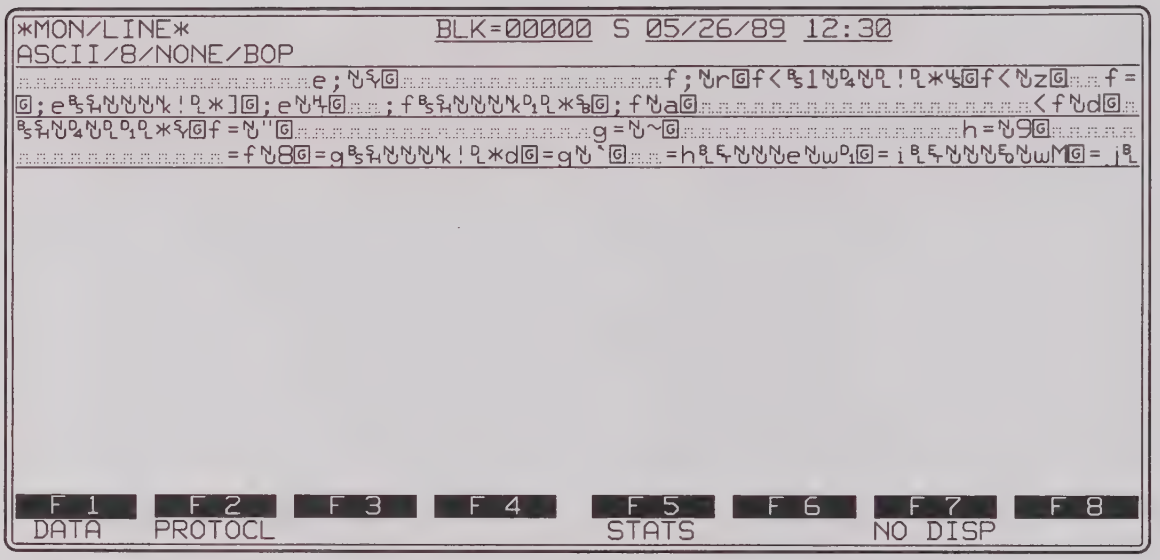


Figure 45-1 SS#7 Dual–line data display, with 7E flags suppressed and redundant LSSUs and FISUs compressed.

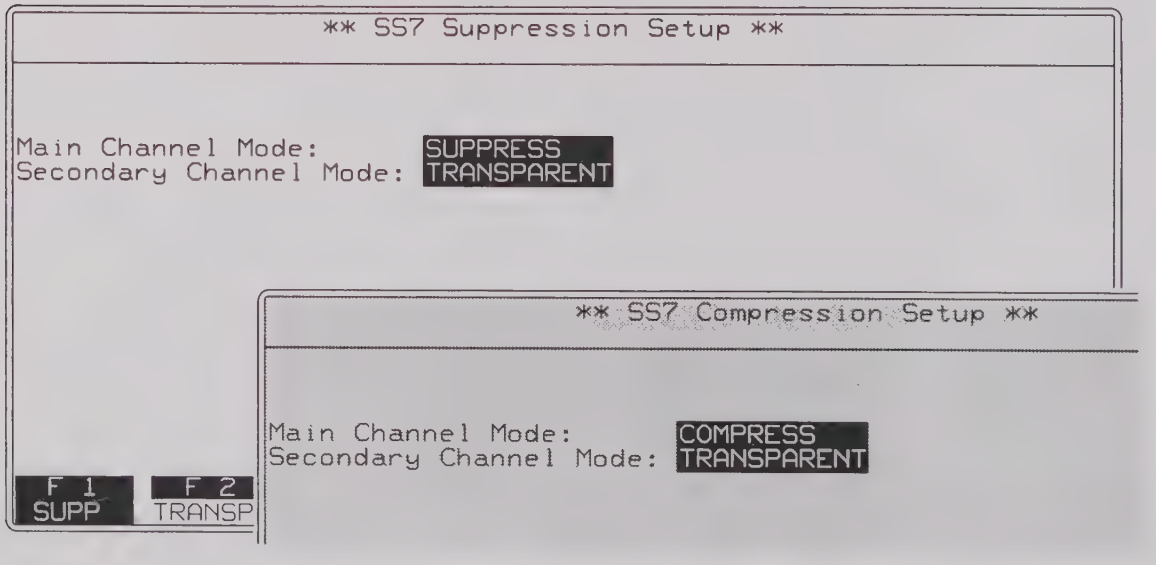


Figure 45-2 Configuration screens for the SS#7 Suppression and SS#7 Compression personality packages for Layer 1.

45 SS#7 Layer 1

SS#7 (CCSS#7) is an abbreviation for the CCITT–defined Common Channel Signalling System #7. The INTERVIEW 8000 Series provides modifiable data display at Layer 1 and special displays for Layer 2 (link) protocol and for Layer 3 (network management) protocol. Automatic selections for SS#7 protocol appear at Layers 2 and 3 of the Protocol Spreadsheet when the correct protocol packages are loaded.

45.1 SS#7 Run–Time Displays

(A) Data Display

Figure 45-1 shows SS#7 data being monitored as dual–line data. 7E flags have been suppressed on the Line Setup screen and Fill–In Signal Units (FISUs) and Link Status Signal Units (LSSUs) have been compressed. As with all dual–line displays, TD and RD data appear on alternate lines and the RD data is underlined. Time fill characters maintain the timing relationships between RD and TD characters, accurate to within one character on the display.

(B) SS#7 Layer Traces

An SS#7 Layer 2 trace is available when SS#7 protocol is loaded at Layer 2. A Layer 3 trace is available when SS#7 is loaded at Layer 3. If you select PROTOCOL and **Layer:** 2 or 3 as the display, this trace will be active when you press RUN. Refer to Sections 46 and 47 for more information on and examples of these traces.

45.2 Setup for SS#7

Three steps are involved in setting up for SS#7 protocol. First, load the desired protocol from the Layer Setup screen as described in Section 8. Then select the correct **Mode** and data **Format** on the Line Setup menu, and, finally, select the options you prefer on the Display Setup menu.

(A) Layer Setup

SS#7 packages are available at Layers 1, 2, and 3. The Layer 1 packages provide data compression or suppression and are described at the end of this section. Options available with the Layer 2 and the Layer 3 packages are discussed in Sections 46 and 47, respectively.

(B) Line Setup

The **Format** selection on the Line Setup menu should be BOP (Bit–Oriented Protocol) when SS#7 is analyzed. This and other Line Setup selections are described in Section 5.

(C) Display Setup

Select initial display options on the Display Setup menu. All available display modes are applicable to SS#7 data. Protocol displays specific to SS#7, which appear only when an SS#7 Personality Package has been loaded, are described in Sections 46 and 47. Data, Display Window, and Program Trace displays are described in Section 6.

You have the option of suppressing 7E flags on the Display Setup menu. See Section 6.3(C).

Data suppressed from the display is available for triggering in real–time. However, it is not available for triggering when character–oriented data from the screen buffer is played back.

It is also possible to suppress all occurrences of a particular type of frame from the display. Refer to the description of the SUPPRESS action in Section 46.

45.3 SS#7 Compression at Layer 1

When the Layer 1 package SS7_CMPRESN is loaded, redundant FISUs and LSSUs are suppressed from the character display and also the Layer 2 trace display in Run mode. That is, only the first in a series of identical Link Status or Fill–In Signal Units is displayed and presented to the trigger program. Subsequent identical units are compressed, until a different type of signal unit is transmitted on the same side of the line.

When you are operating the INTERVIEW in a dual–port environment, you may configure the unit to compress the data on the main channel only, the secondary channel only, or on both channels. A configuration screen is accessible when the SS7_CMPRESN Layer 1 package is loaded; press PROTSEL, LAYER–1 (see Figure 45-2). You may designate TRANSPARENT or COMPRESS for each channel.

> **NOTE:** When running an application program such as *GSM Dual–Protocol Trace (LAPD/SS#7 MTS), OPT–951–106–1,* and monitoring SS#7 data on one port, the user may wish to compress or suppress the data on the other port.

When you are operating in a single–port environment, the choice you make in the **Main Channel Mode** field will be utilized for your data at all times. If you choose TRANSPARENT in this field, *data compression will not take place*. The selection in the **Secondary Channel Mode** field is ignored when DUAL PORT is not selected (or not an option) on the TIM Setup screen.

NOTE: Additional overhead is added when this personality package is loaded, causing some performance degradation. This occurs with *either* choice on the Setup screen. If you are operating in a single−port environment and *do not* wish data compression to occur, we recommend that you load NO PACKAGE at Layer 1 rather than simply choose **Main Channel Mode:** TRANSPARENT on the Setup screen.

Compare Figure 45-3 to Figure 45-4 (in which FISUs and LSSUs have been compressed).

NOTE: The number of suppressed signal units and flags can be monitored via C variables. Refer to Table 78-1.

Bit−image recording of data is not affected by Layer 1 compression. Simply select and load NO PACKAGE at Layer 1 and play the same bit−image data back again, to cancel the effects of compression. All the original Link Status and Fill−In Signal Units will be presented to the screen display and the triggers.

45.4 SS#7 Suppression at Layer 1

The Layer 1 SS7_SUPP package is similar to the compression package described above. The suppression package, however, passes only Message Signal Units (MSUs) up to Layer 2. Link Status Signal Units are suppressed and all Fill−In Signal Units are suppressed, except the first one in which the value of the BIB has been inverted.

A similar configuration screen is present for channel setup in a dual−port environment when the SS7_SUPP Layer 1 package is loaded (see Figure 45-2).

NOTE: When running an application program such as *GSM Dual−Protocol Trace (LAPD/SS#7 MTS), OPT−951−106−1,* and monitoring SS#7 data on one port, the user may wish to suppress or compress the data on the other port.

The suppression package has one associated C variable, *ss7_compression_mode.* When the value of this variable is zero, FISUs are suppressed; when its value is non−zero, FISUs are compressed, exactly as in the SS#7 compression package.

MON/LINE BLK=00000 S 05/26/89 12:04
ASCII/8/NONE/BOP

SRC	TYPE	LI	BIB	BSN	FIB	FSN	TIME	BCC
DTE	FILL-IN	00	1	4D	1	3F	1204:31.257	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.280	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.303	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.327	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.346	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.375	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.395	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.423	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.441	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.461	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.515	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.546	G
DTE	FILL-IN	00	1	4D	1	3F	1204:31.577	G
DCE	MESSAGE	08	1	3F	1	4E	1204:31.598	G
DTE	FILL-IN	00	1	4D	1	3F	1204:32.019	G

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8
L2TRACE L3TRACE

Figure 45-3 All LSSUs and FISUs detected are displayed on this screen.

MON/LINE BLK=00000 S 05/26/89 12:04
ASCII/8/NONE/BOP

SRC	TYPE	LI	BIB	BSN	FIB	FSN	TIME	BCC
DTE	FILL-IN	00	1	4D	1	3F	1204:31.222	G
DCE	FILL-IN	00	1	3F	1	4D	1204:31.576	G
DCE	MESSAGE	08	1	3F	1	4E	1204:31.593	G
DCE	FILL-IN	00	1	3F	1	4E	1204:32.021	G
DTE	FILL-IN	00	1	4E	1	3F	1204:32.048	G
DTE	MESSAGE	08	1	4E	1	40	1204:32.069	G
DCE	FILL-IN	00	1	40	1	4E	1204:32.080	G
DTE	FILL-IN	00	1	4E	1	40	1204:32.115	G
DTE	MESSAGE	08	1	4E	1	41	1204:32.130	G
DCE	FILL-IN	00	1	41	1	4E	1204:32.154	G
DTE	FILL-IN	00	1	4E	1	41	1204:32.177	G
DCE	MESSAGE	08	1	41	1	4F	1204:32.195	G
DCE	FILL-IN	00	1	41	1	4F	1204:32.208	G
DTE	FILL-IN	00	1	4F	1	41	1204:32.237	G
DCE	MESSAGE	08	1	41	1	50	1204:32.271	G

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8
L2TRACE L3TRACE

Figure 45-4 Redundant LSSUs and FISUs have been compressed in this data.

46 SS#7 Layer 2

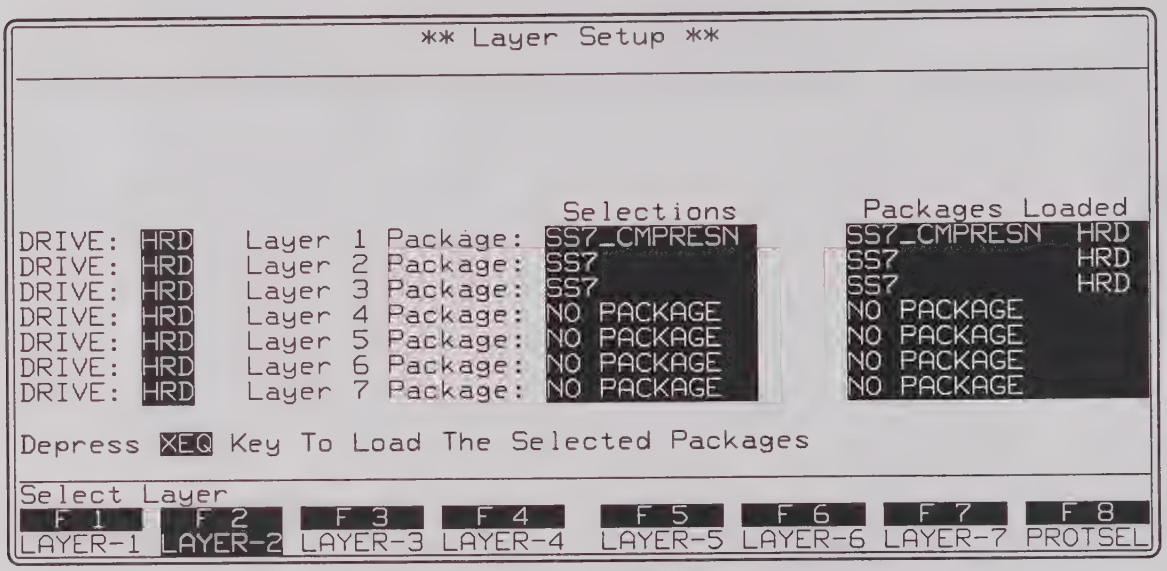


Figure 46-1 The SS#7 personality package for Layer 2 is loaded from the Layer Setup screen.

46 SS#7 Layer 2

SS#7 (CCSS#7) is an abbreviation for the CCITT–defined Common Channel Signalling System #7. At Layer 2, SS#7 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 46-1 shows the Layer Setup screen configured to load in the Layer 2 SS#7 package from the hard disk. Refer to Section 8 for details on operating the Layer Setup screen.

46.1 Set Up for SS#7

(A) Layer Setup

The SS#7 package at Layer 1 allows you to compress redundant frames, as explained in Section 45.

The Layer 2 SS#7 package consists of the following:

- A protocol trace (illustrated in Figure 46-2) that distills from SS#7 data the Level 2 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate SS#7 programming. Figure 46-4 shows the softkey path to the first rack of condition softkeys when the SS#7 package is loaded in at Layer 2.

(B) Line and Display Setup

Be sure that the **Format** selection is BOP (for Bit–Oriented Protocol) when you are testing SS#7. Select display options on the Display Setup menu. These options are discussed in Section 45.2.

46.2 Protocol Trace

The Layer 2 SS#7 package includes an automatic frame–trace display that summarizes link–level activity. This trace mode is enabled whenever the unit is in Run mode, both real–time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE to bring the protocol trace for SS#7 Layer 2 to the screen. (If the SS#7 package for Layer 3 is also loaded in, the L2TRACE softkey will appear after you have pressed PROTOCL, F2 on the primary rack of display–mode softkeys.)

When running in High–Speed Frame Mode, more data could be passed to Layer 2 than there is room for in the buffer; this will cause an FEB overrun. If this happens, the error message **FE Buffer Overflowed – Some Frames Lost** will appear on the prompt line. The first time an FEB overrun occurs, an audible alarm will also sound; subsequent reccurrences will cause only the message to display (without any alarm). The trace will restart again but some data is lost with each occurrence.

Figure 46-2 is an example of the Layer 2 trace display. Each horizontal row in the trace represents a frame.

MON/LINE BLK=00000 S 05/26/89 12:30
ASCII/8/NONE/BOP

SRC	TYPE	LI	BIB	BSN	FIB	FSN	TIME	BCC
DCE	MESSAGE	08	1	3B	1	66	1229:59.160	G
DCE	FILL-IN	00	1	3B	1	66	1229:59.203	G
DTE	FILL-IN	00	1	66	1	3B	1229:59.242	G
DTE	MESSAGE	08	1	66	1	3C	1229:59.278	G
DTE	FILL-IN	00	1	66	1	3C	1229:59.306	G
DCE	FILL-IN	00	1	3C	1	66	1229:59.352	G
DTE	MESSAGE	08	1	66	1	3D	1229:59.371	G
DTE	FILL-IN	00	1	66	1	3D	1229:59.421	G
DCE	FILL-IN	00	1	3D	1	66	1229:59.441	G
DCE	MESSAGE	08	1	3D	1	67	1229:59.470	G
DCE	FILL-IN	00	1	3D	1	67	1229:59.506	G
DTE	FILL-IN	00	1	67	1	3D	1229:59.521	G
DCE	MESSAGE	07	1	3D	1	68	1229:59.559	G
DTE	FILL-IN	00	1	68	1	3D	1229:59.588	G
DCE	MESSAGE	07	1	3D	1	69	1230:50.007	G

F 1 L2TRACE F 2 L3TRACE F 3 F 4 F 5 F 6 F 7 F 8

Figure 46-2 SS#7 Layer 2 Protocol Trace.

(A) The Protocol Trace in Freeze Mode

Press [FREEZE] to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing [ROLL FWD] or [↓] moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing [ROLL BACK] or [↑] moves the viewing window up to add a line of older data to the top of the screen.

Depression of the [NEXT PAGE] key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of [PREV PAGE] adds 15 lines of older frames.

The frame displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. Compare Figure 46-2 with Figure 46-3. To view the raw data that generated a particular line in the trace display,

use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys. Figure 46-3 shows part of a dual−line data screen in Freeze mode. The first frame in the display is the same one that is traced at the top of Figure 46-2.

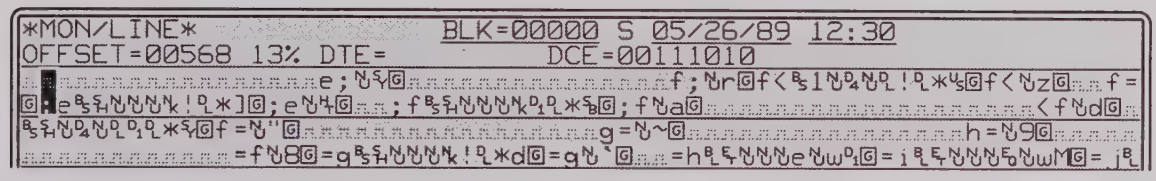


Figure 46-3 Data−display of Protocol Trace shown in Figure 46-2.

(B) Trace Columns

The columns in the protocol trace for Layer 2 SS#7 are explained below.

1. *Source.* The **SRC** column identifies the lead on which the signal unit was monitored, TD (DTE) or RD (DCE). Just as on the data−display, RD data is underlined.

2. *Type.* The second column, **TYPE**, lists the signal−unit type [STATUS= for a Link Status Signal Unit (LSSU), FILL_IN for a Fill−In Signal Unit (FISU), or MESSAGE for a Message Signal Unit (MSU)]. For an LSSU, the status type is also given as an abbreviation.

   The format for FISUs and LSSUs are shown at the end of this section, in Figure 46-10 and Figure 46-11. Abbreviations and values for Link Status Signal Unit types are defined in Table 46-1.

Table 46-1
LSSU Status Field
(Bits 2−0 of first Status Field Octet)

Acronym	Definition	Binary Value
O	Out of Align	000
N	Normal	001
E	Emergency	010
OS	Out of Service	011
PO	Processor Outage	100
B	Busy	101

Note: Bits 7−3 of the First Status Field Octet (SFO) are spare.

A second Status Field Octet may be present.

3. *Length indicator.* The value of the Length Indicator byte (**LI**) is given in the third column of the display. A value of 00 here indicates a FISU, a value of 01 or 02 indicates an LSSU, and a hex value of 03–3F indicates an MSU.

4. *Backward indicator bit.* The fourth column, labeled **BIB**, provides the value of the Backward Indicator Bit.

5. *Backward sequence number.* The hex value of the Backward Sequence Number (**BSN**) is listed in the next column.

6. *Forward indicator bit.* The **FIB** column provides the Forward Indicator Bit.

7. *Forward sequence number.* The Forward Sequence Number (**FSN**) is displayed in hex in the next column.

8. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

   When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

   - For bit–image data

     Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
     Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

   - For character data

     Data To Record: CHARACTER
     Time Ticks: ON

     or

     Data To Record: CHARACTER
     Time Ticks: OFF
     Frame Timestamps: YES (Front–End Buffer Setup screen; see Section 9)
     Timestamp Resolution: 1mS or 1uS

   If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

9. *Frame checking.* An SS#7 frame ends as soon as a $^{7}\varepsilon$ flag or seven 1–bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

■ for abort is posted to the displays when a frame is ended by seven 1−bits.

46.3 Monitor Conditions

When the Layer 2 SS#7 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 46-4.

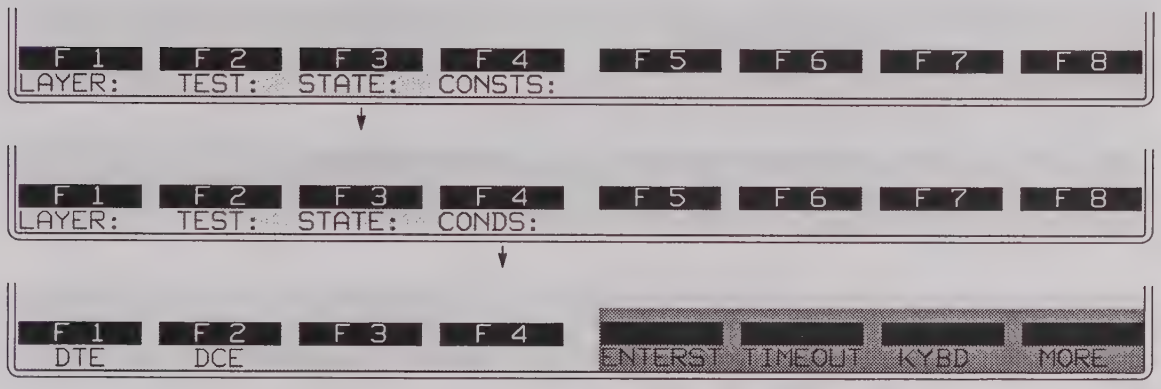


Figure 46-4 To monitor line conditions, first select *DTE* or *DCE*.

(A) Signal−Unit Types

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represents *types* of Signal Units.

1. *Fill−In and Message.* The softkeys for FILL_IN (Fill−In Status Unit), MESSAGE (Message Signal Unit), and STATUS= (Link Status Signal Unit) are illustrated in Figure 46-5. Press a softkey to write one of these signal−unit types to the Layer 2 spreadsheet.

   DTE (or DCE) FILL_IN and DCE (or DTE) MESSAGE are complete conditions and will come true if a matching frame is monitored. BIB, FIB, and BCC conditions may be added to the simple frame mnemonic, but they are optional.

2. *Link Status.* DTE (or DCE) STATUS= is not a complete condition. Select an LSSU type from the third softkey rack shown in Figure 46-5. The full set of abbreviations and their meanings is given in Table 46-1.

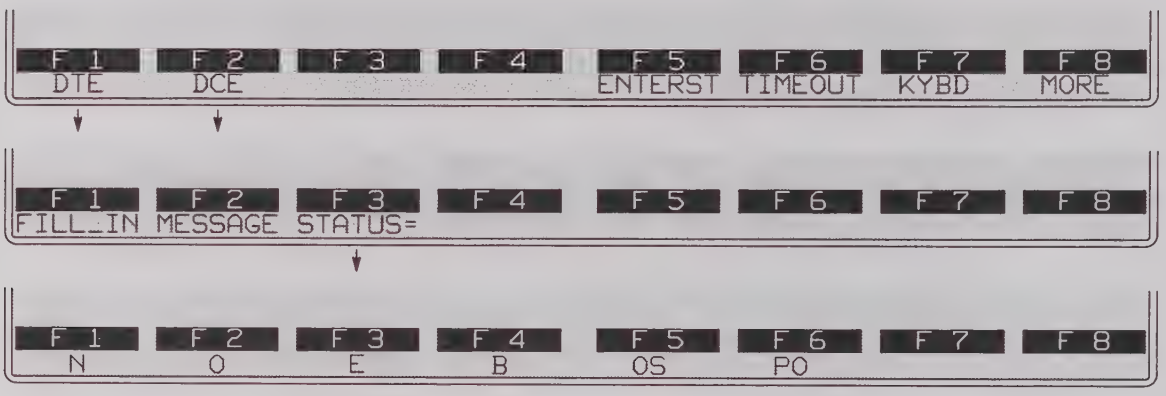


Figure 46-5 Frame types.

(B) Forward and Backward Indicator Bits

For any SS#7 frame type, you have the option of specifying the value of the Backward Indicator Bit (BIB= 0, BIB= 1) and the value of the Forward Indicator Bit (FIB= 0, FIB= 1):

```
CONDITIONS: DTE FILL_IN BIB= 1 FIB= 1
```

The softkey path to BIB and FIB is shown in Figure 46-6. If you omit either the FIB or the BIB field, the omitted FIB or BIB is not checked in the received frame. Press [DONE] to bypass the BIB and FIB conditions.

To make BIB and FIB selections for Link Status Signal Units, follow the softkey path shown in Figure 46-7.

(C) BCC Conditions

For any SS#7 frame type, you also have the option of specifying a BCC or abort condition:

```
CONDITIONS: DTE MESSAGE BD_BCC
```

The softkey path to BCC selections for Fill–In and Message Signal Units is shown in Figure 46-6. Press [DONE] to bypass the BCC condition.

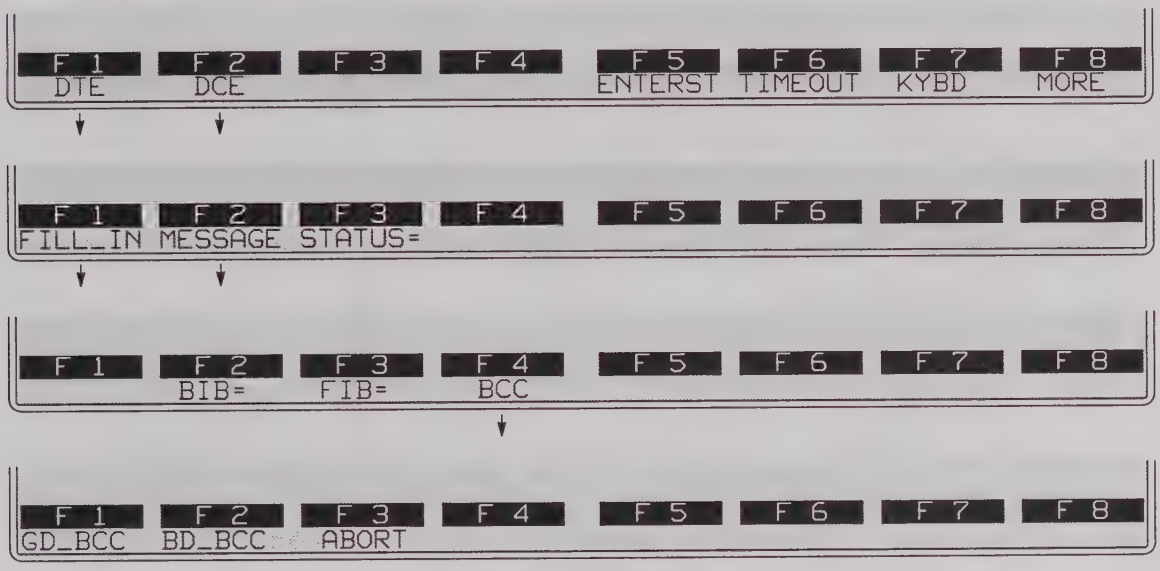


Figure 46-6 You may specify *BIB, FIB,* or *BCC* conditions immediately following *FILL_IN* or *MESSAGE* selections.

To make BCC selections for Link Status Signal Units, follow the softkey path shown in Figure 46-7.

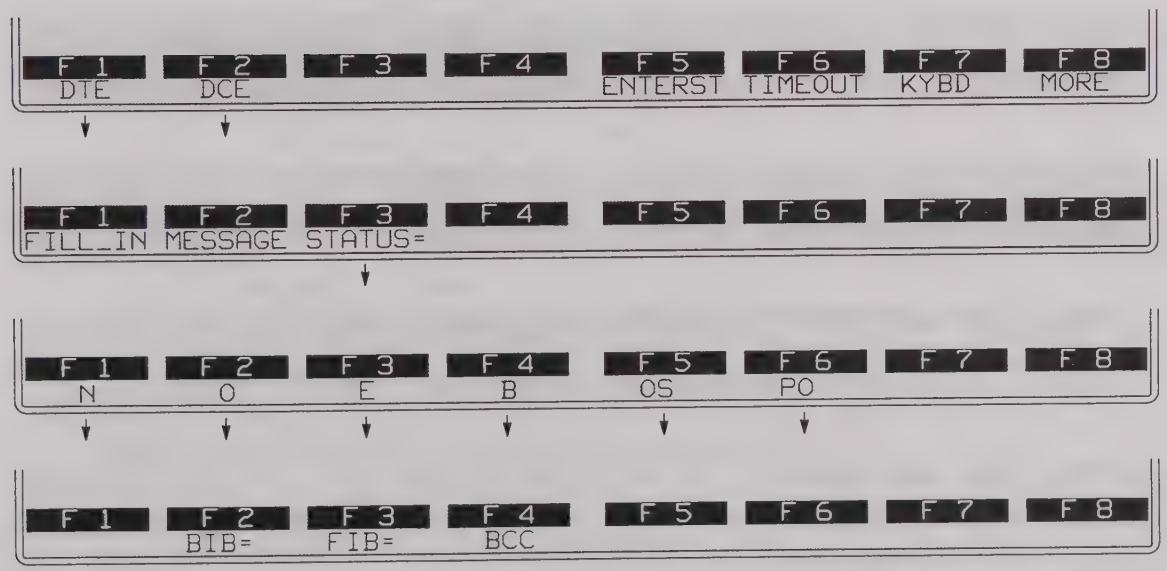


Figure 46-7 You must specify the type of Link Status Signal Unit for a *STATUS=* spreadsheet condition. *BIB, FIB,* and *BCC* conditions may then be selected.

46.4 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 2 protocol trace (see Section 46.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE and DCE conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE or DCE condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the SS#7 Layer 2 protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F1 on the first rack of action softkeys. Figure 46-8 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

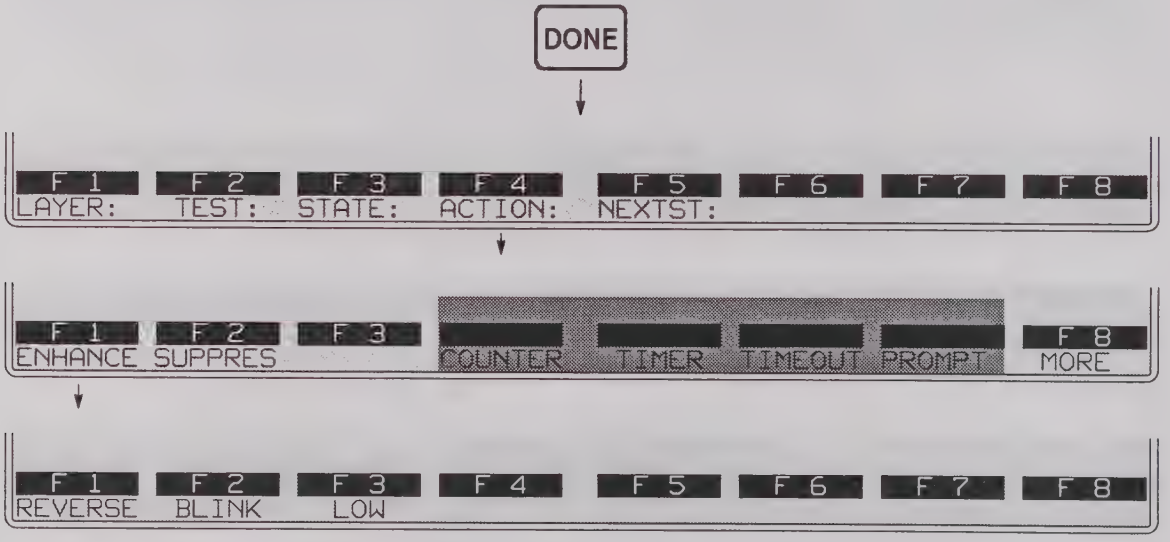
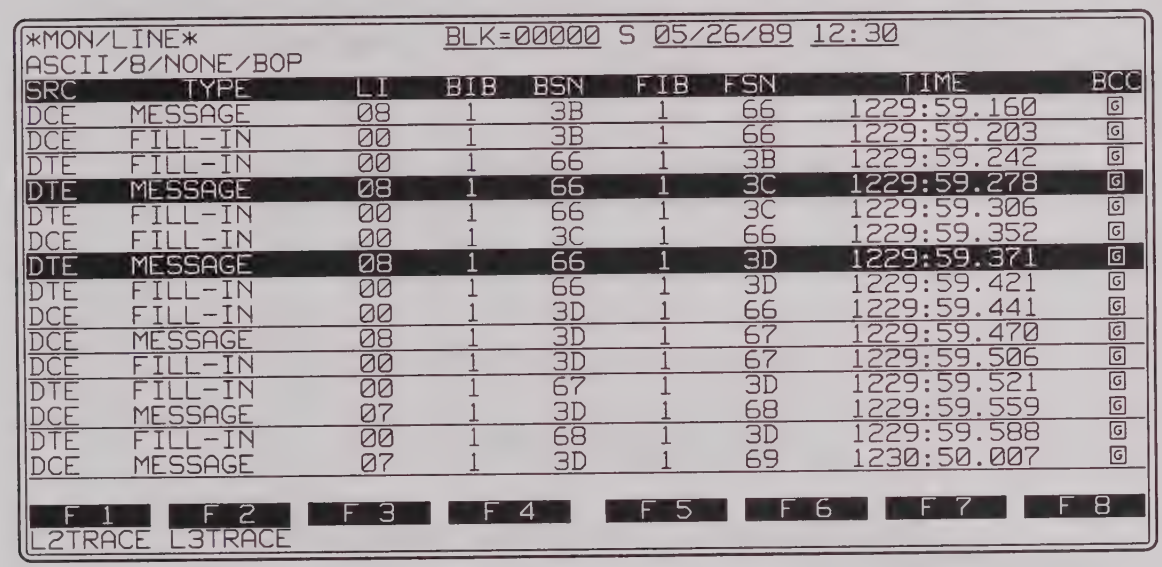


Figure 46-8 Selected frames on the protocol trace may be enhanced or suppressed.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.



MON/LINE BLK=00000 S 05/26/89 12:30
ASCII/8/NONE/BOP

SRC	TYPE	LI	BIB	BSN	FIB	FSN	TIME	BCC
DCE	MESSAGE	08	1	3B	1	66	1229:59.160	G
DCE	FILL-IN	00	1	3B	1	66	1229:59.203	G
DTE	FILL-IN	00	1	66	1	3B	1229:59.242	G
DTE	MESSAGE	08	1	66	1	3C	1229:59.278	G
DTE	FILL-IN	00	1	66	1	3C	1229:59.306	G
DCE	FILL-IN	00	1	3C	1	66	1229:59.352	G
DTE	MESSAGE	08	1	66	1	3D	1229:59.371	G
DTE	FILL-IN	00	1	66	1	3D	1229:59.421	G
DCE	FILL-IN	00	1	3D	1	66	1229:59.441	G
DCE	MESSAGE	08	1	3D	1	67	1229:59.470	G
DCE	FILL-IN	00	1	3D	1	67	1229:59.506	G
DTE	FILL-IN	00	1	67	1	3D	1229:59.521	G
DCE	MESSAGE	07	1	3D	1	68	1229:59.559	G
DTE	FILL-IN	00	1	68	1	3D	1229:59.588	G
DCE	MESSAGE	07	1	3D	1	69	1230:50.007	G

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8
L2TRACE L3TRACE

Figure 46-9 DTE Message Signal Units have been enhanced.

Figure 46-9 shows one screen of a Layer 2 protocol trace in which DTE MSUs have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

```
CONDITIONS: DTE MESSAGE
ACTIONS:  ENHANCE REVERSE
```

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 46-8 shows the softkey path to SUPPRES.

46.5 SS#7 Emulation

You may use SS#7 protocol for interactive testing of a DTE or a DCE. However, transmitted strings must be entered manually, since no automatic selections are currently available for emulation, either at Layer 2 or Layer 3.

46.6 SS#7 Frame Structures and Values

The format for Fill–In Signal Units is given in Figure 46-10.

The format for Link Status Signal Units is given in Figure 46-11. Table 46-1 lists possible LSSU status values.

Refer to Section 47 for the structure of a Message Signal Unit and for the differences in CCITT and ANSI (US Standard) frame format. Also, consult Section 47 for any information pertinent to SS#7 Level 3.

NOTE: Frame format, unless otherwise stated, reflects the frame as displayed on the screen, not the actual transmission order.

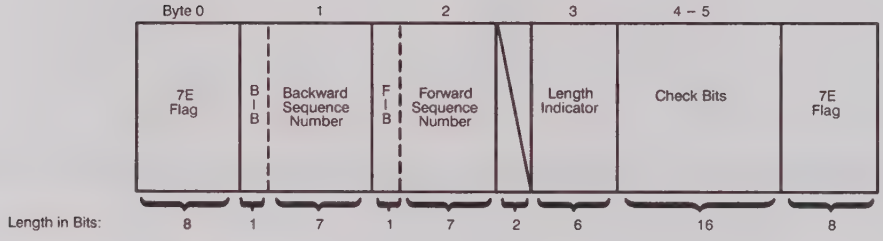


Figure 46-10 Format of Fill–in Signal Unit (FISU).

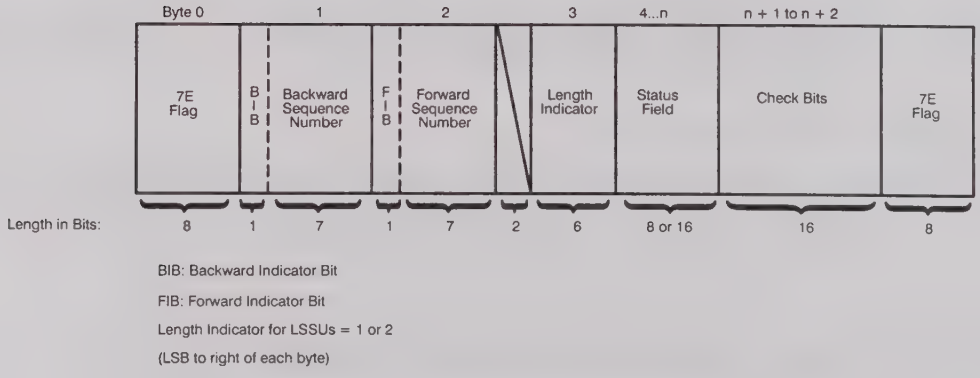


Figure 46-11 Format of Link Status Signal Unit (LSSU).

47 SS#7 Layer 3

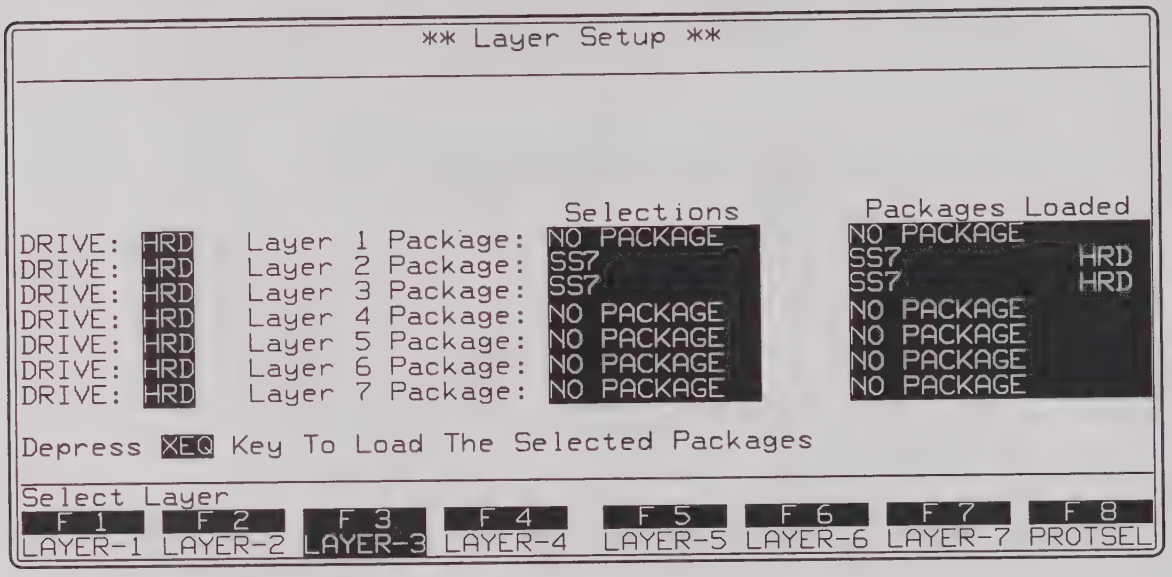


Figure 47-1 The SS#7 personality package for Layer 3 is loaded from the Layer Setup screen.

47 SS#7 Layer 3

SS#7 (CCSS#7) is an abbreviation for the CCITT–defined Common Channel Signalling System #7. At Layer 3, SS#7 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 47-1 shows the Layer Setup screen configured to load in the Layer 3 SS#7 package from the hard disk. Refer to Section 8 for details on operating the Layer Setup screen.

The Layer 3 SS#7 package consists of the following:

- A special SS#7 Packet Level Setup screen that controls certain parameters when the unit is tracing SS#7.
- A protocol trace (illustrated in Figure 47-2) that distills from SS#7 data the Layer 3 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 3 on the Protocol Spreadsheet that facilitate SS#7 programming. Figure 47-3 shows the softkey path to the first rack of condition softkeys when the SS#7 package is loaded in at Layer 3.

47.1 Packet–Level Setup

The SS#7 Packet Level Setup screen must be configured correctly for an accurate trace display.

To bring up this screen, first go to the Layer Setup screen (press PROGRAM, F5). Execute the SS#7 selection at Layer 3: **SS7** should appear in the **Packages Loaded** column. Press F8 (labeled PROTSEL) to bring up the prompt to **Select Protocol Configuration Screen**. Then press F3 (LAYER–3) to call up the SS#7 Packet Level Setup screen.

The only parameter field on this screen is **National Format**. This field allows you to specify the format of the point codes in the label portion of a frame with a National Network Indicator (binary 10 or 11). Select CCITT or ANSI (US Standard). It is important that you select the correct format prior to testing, since the format which the INTERVIEW anticipates for national frames differs depending on your choice. Compare the two frame structures in Figure 47-9.

(A) CCITT Format

When you select CCITT, national frames are structured like international frames. These frames contain 14–bit point codes within a 32–bit routing label.

> **NOTE:** The INTERVIEW represents a 14–bit CCITT point code as two hexadecimal bytes, and it pads the two most significant bit positions of the left–most byte with zeros.

(B) ANSI Format

The default selection is ANSI, which is US Standard format, in which national frames contain 24–bit point codes within a 56–bit routing label.

47.2 Protocol Trace

While the unit is in Run mode, press the softkey for PROTOCL (F2 on the primary rack of display–mode softkeys) and then the softkey for L3TRACE (F2)to bring the protocol trace for SS#7 Layer 3 to the screen. Figure 47-2 is an example of this trace display. Each horizontal row in the trace represents an MSU.

(A) The Protocol Trace in Freeze Mode

Press FREEZE to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing ROLL FWD or ↓ moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing ROLL BACK or ↑ moves the viewing window up to add a line of older data to the top of the screen.

Depression of the NEXT PAGE key adds 15 lines—one full page—of newer frames to the frozen trace screen. Depression of PREV PAGE adds 15 lines of older frames.

The MSU displayed on the top line of frozen trace–data will appear as the first frame in the raw–data or data–plus–leads display. To view the raw data that generated a particular line in the trace display, use ROLL FWD or ROLL BACK (or ↑ or ↓) to move the line in question to the top of the screen. Then press one of the data softkeys.

(B) Trace Columns

There are eleven columns in the Layer 3 display.

1. *Source.* The first column (**SRC**) identifies the MSU source as DTE (TD) or DCE (RD). Just as on the data–display, RD data is underlined.

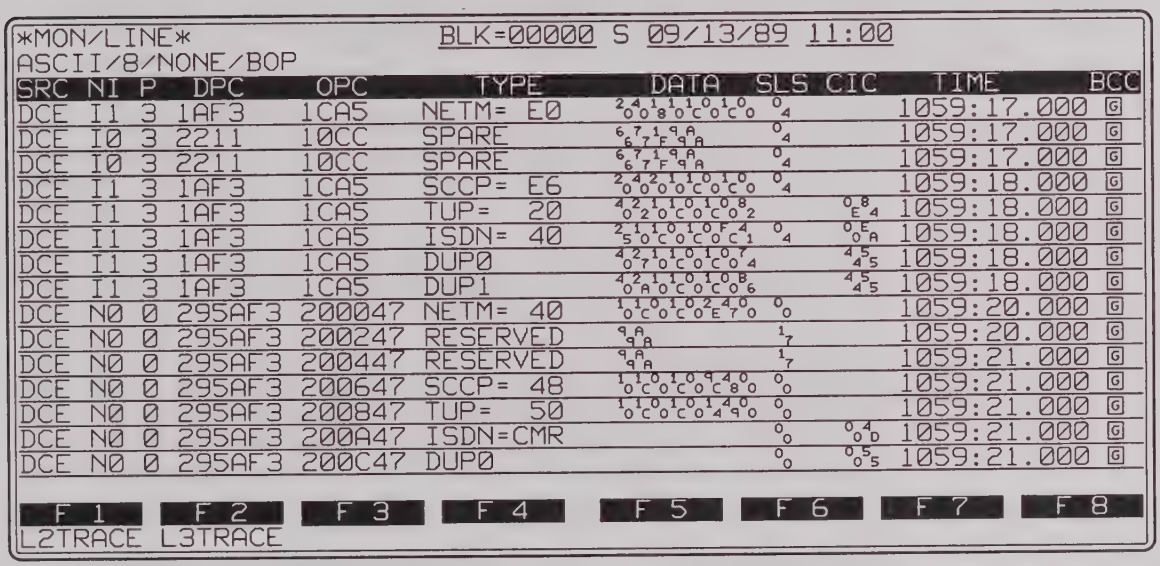


Figure 47-2 SS#7 Layer 3 Protocol Trace.

2. *Network indicator.* The second column (labeled **NI**) interprets the Network Indicator in a two–character field. The first bit in the two–bit network Indicator is always set to 0 in international signal units; the first bit is set to 1 in all national signal units. On the SS#7 trace, the bit value is represented as an "I" to indicate CCITT international format or "N" to represent national format, whether ANSI or CCITT. The value of the second bit in the Network Indicator is also displayed on the trace screen.

3. *Priority.* The third column (**P**) displays the value of the Priority code, regardless of format. The Priority code has a value of 0–3 for ANSI (US format). These bits are spare in CCITT International or National format.

4. *Destination point code.* The next column (**DPC**) provides the Destination Point Code. When the **NI** field indicates ANSI format (N), three hex bytes are used in the **DPC** column. CCITT format (I) allocates only 14 bit positions. The 14 bits are displayed as two bytes, with the two left–most bit positions padded with zeros.

5. *Originating point code.* The next column (**OPC**) provides the Originating Point Code. When the **NI** field indicates ANSI format (N), three hex bytes are used in the **OPC** column. CCITT format (I) allocates only 14 bit positions. The 14 bits are displayed as two bytes, with the two left–most bit positions padded with zeros.

6. *Type.* The **TYPE** column defines the message type (that is, the Service Indicator) for MSUs. The values of the different Service Indicators are given in Table 47-1. For Integrated Services Data Network User Part (ISUP), Network Management (NETM), Telephone User Part (TUP), or Signalling Connection Control Part (SCCP) messages, the header type is given either as an abbreviation or as a hex value. Abbreviations are defined in Table 47-2 through Table 47-5. A hex value appears when the header has no defined abbreviation.

7. *Data.* The **DATA** column displays up to eight bytes of data in hexadecimal format. The amount of additional data displayed is defined for each message type as shown in Table 47-2 through Table 47-5.

8. *Signalling link selection.* The **SLS** column gives the hex value (one or two bytes) of the signalling link selection when an SLS is present. Depending on the MSU type, the SLS occupies four or five bits within the MSU. The 4– or 5–bit SLS is always shown as a hex byte in the Layer 3 trace. The three or four remaining bit positions of the byte are padded with zeros.

9. *Circuit identification code.* When a Circuit Identification Code is present, it is listed in the next column. The **CIC** is a 16–bit field within the MSU and is represented on the trace screen as two hex characters.

10. *Time.* The time of the arrival of the *end of the MSU* at the DTE or DCE monitor is provided by a 24–hour clock and posted to the trace display. The clock is accurate to the milli– or microsecond.

    When time values are incorporated in data, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds. To incorporate time values into recorded data, make the following selections:

    - For bit–image data

      Data To Record: BIT IMAGE (Record Setup menu; see Section 7)
      Time Ticks: ON (Front–End Buffer Setup screen; see Section 9)

    - For character data

      Data To Record: CHARACTER
      Time Ticks: ON

      or

Data To Record: CHARACTER
Time Ticks: OFF
Frame Timestamps: YES (Front−End Buffer Setup screen; see Section 9)
Timestamp Resolution: 1mS or 1uS

If time values are not incorporated in data during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

11. *Frame checking.* An SS#7 frame ends as soon as a $^{7}\varepsilon$ flag or seven 1−bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol G denotes a good frame check, while B symbolizes a bad frame.

A for abort is posted to the displays when a frame is ended by seven 1−bits.

NOTE: In MSUs which are incomplete, the header is expanded if sufficient information is present. Additional unexpanded bytes displayed may include the first FCS byte followed by FF bytes which pad to the end of the frame display.

47.3 Monitor Conditions

When the Layer 3 SS#7 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 3. See Figure 47-3.

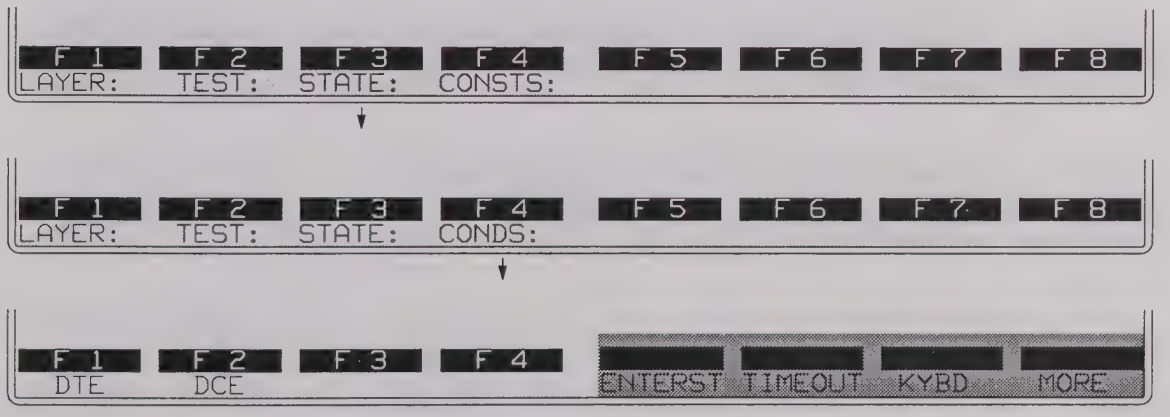


Figure 47-3 To monitor line conditions, first select *DTE* or *DCE*.

(A) Message Signal Unit Types

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represents *types* of MSUs. See Figure 47-4.

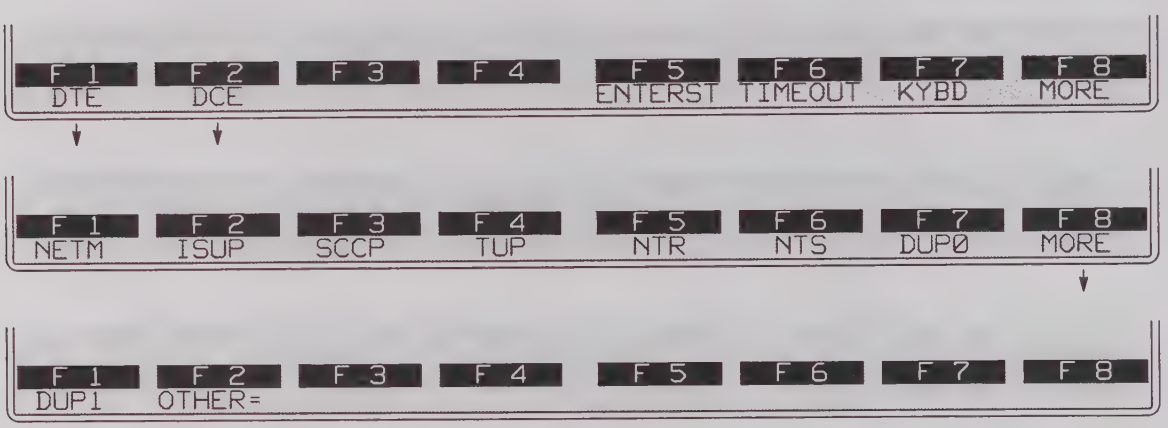


Figure 47-4 MSU types.

The Message Signal Units which appear as automatic selections include NETM (Network Management), ISUP (Integrated Services Data Network User Part), SCCP (Signalling Connection Control Part), TUP (Telephone User Part), NTR (Network Test Regular), NTS (Network Test Special), DUP0 or DUP1 (Data User Part 0 and Data User Part 1), and OTHER (for user–specified MSUs). Values for the different MSU types are given in Table 47-1.

Press a softkey to write one of these MSU–types to the Layer 3 spreadsheet. DTE or DCE followed by an MSU–type mnemonic—DTE NETM, for example, or DCE ISUP—is a complete condition and will come true if a matching MSU is monitored.

When you select OTHER as the MSU type, you must specify the value of the four low–order bits for the SIO (Service Indicator Octet). Enter the value of the four low–order bits as a single hexadecimal digit. (Do not use the [HEX] key.)

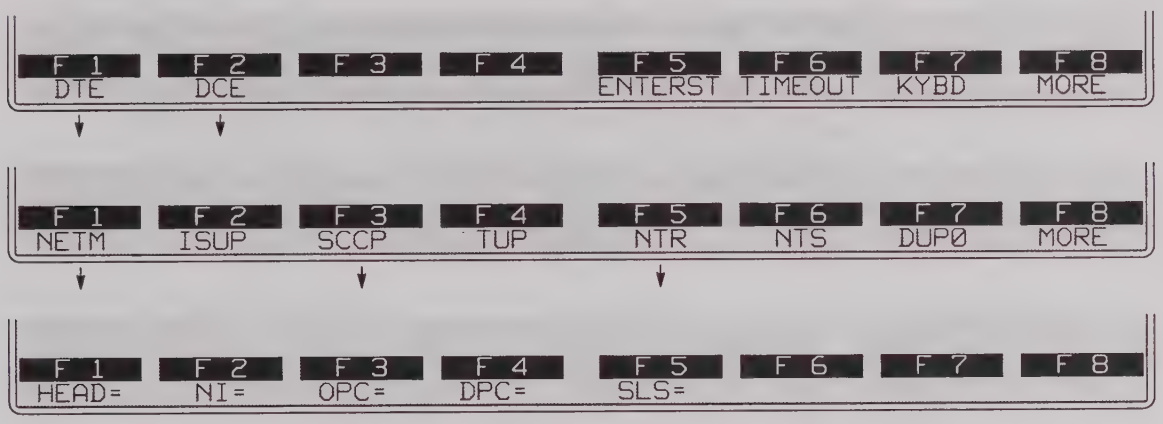


Figure 47-5 Selected MSU type determines subsequent softkey selections.

Certain secondary fields appear for the remaining MSU types. The fields for NETM, SCCP, and NTR selections are shown in Figure 47-5. These optional selections for MSUs permit you to specify Network Identifier, Originating Point Code, Destination Point Code, Signalling Link Selection (where applicable), Circuit Identification Code (where applicable), and Header (where applicable). See the spreadsheet example in Figure 47-6.

```
LAYER: 3
  TEST: 13check
    STATE: charge
      CONDITIONS: DTE SCCP HEAD= CC OPC= 1CA5 DPC= 1AF3
      ACTIONS: ENHANCE BLINK
```

Figure 47-6 Three optional conditions (*HEAD=* , *OPC=* , and *DPC=*) have been selected for SCCP Message Signal Units on the DTE side.

(B) MSU Header

When certain MSUs are selected, you may also specify the header in the MSU. Selectable headers are available when the MSU selection is NETM, NTR, SCCP, TUP, or ISUP. Header selections for NETM, SCCP, TUP, or ISUP MSUs, their meanings, and their values are given in Table 47-2 through Table 47-5. Header selections for NTR MSUs are given in Note 8.

(C) Network Indicator

Once you have specified an MSU type, you may specify the Network type. Select NI= N to signify a National network; select NI= I to indicate an International network.

NOTE: National in this case refers to the value of the first two bits in the Service Indicator Octet. It does NOT distinguish between CCITT National and ANSI (US Standard) format. To designate the appropriate frame structure, you MUST select CCITT or ANSI as a Layer Setup parameter as described in Section 47.1. Legal values in the **OPC** and **DPC** fields and the actual data strings anticipated as CONDITIONS on the Protocol Spreadsheet are determined by your Layer Setup selection for Network type.

(D) Originating Point Code

Type in the **OPC** as hexadecimal digits. (Do not use the [HEX] key.) The size of this field differs for a CCITT MSU or an ANSI (US Standard) MSU.

In CCITT format, the OPC is a 14–bit field. Type in a two–byte entry (four hex digits), and assume that the two high–order bits of the left–most byte are zeros. (Legal values are 0000 to 3FFF.)

In ANSI format, the OPC is a three–byte field. (Legal values are 000000 to FFFFFF.)

(E) Destination Point Code

Type in the DPC as hexadecimal digits. (Do not use the [HEX] key.) As with the OPC, the size of this field varies, depending on the frame structure (CCITT or ANSI) selected as a Layer Setup parameter. Legal entries are the same as for Originating Point Codes.

(F) Signalling Link Selection

MSUs which contain an SLS are listed in Table 47-1.

The SLS occupies four bits in CCITT format. The four high–order bits of the same byte are always zero. Valid entries for CCITT format are 00 to 0F.

In ANSI (US Standard) format, the SLS occupies five low–order bits of a byte, and the three high–order bits of the same byte are set to zero. Valid entries for ANSI format are 00 to 1F.

Enter two hexadecimal digits as the SLS. (Do not press [HEX].)

(G) Circuit Identification Code

MSUs which contain a CIC are listed in Table 47-1. For ISUP MSUs, the CIC is a two–byte field. Enter the CIC as hexadecimal digits in the range 0000 to FFFF. Do not use the [HEX] key. For TUP MSUs, enter up to three hexadecimal digits in the range 000 to FFF.

47.4 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 3 protocol trace (see Section 47.2). They do not suppress and enhance data on the raw–data display. Raw data is enhanced and suppressed at Layer 1.

DTE and DCE conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE or DCE condition comes true at Layer 3, the MSU that satisfied the condition can be enhanced on the SS#7 Layer 3 protocol–trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—F1 on the first rack of action softkeys. Figure 47-7 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

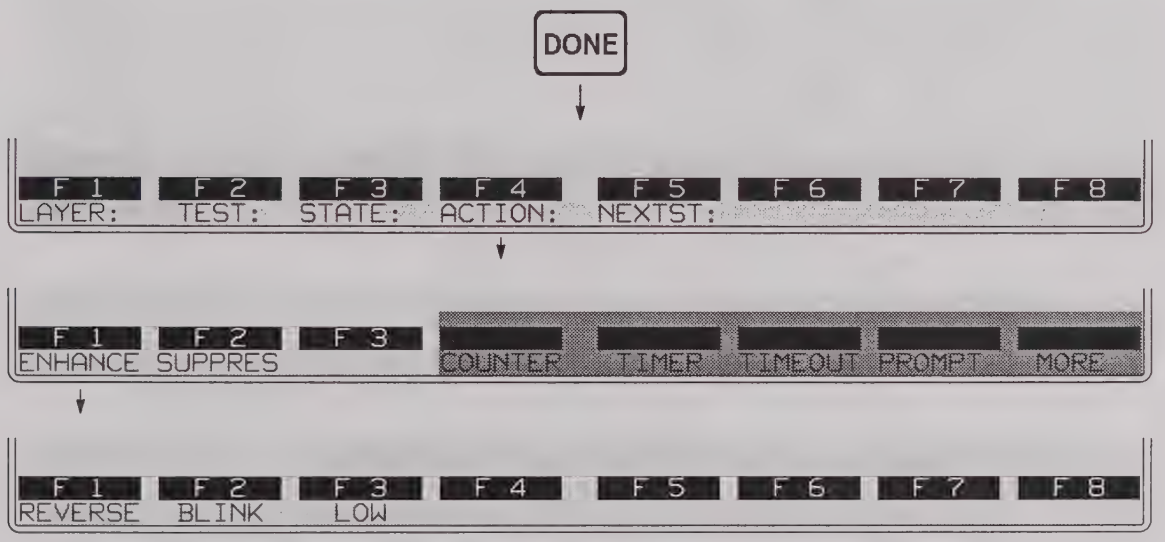


Figure 47-7 Selected MSUs on the protocol trace may be enhanced or suppressed.

Reverse–image and blink enhancements affect the plasma–display screen. In addition, a low–intensity enhancement can be applied to screens that are transmitted to a black–and–white monitor connected at the RS–170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 18.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

Figure 47-8 shows one screen of a Layer 3 protocol trace in which NETM MSUs have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DCE NETM
ACTIONS: ENHANCE REVERSE

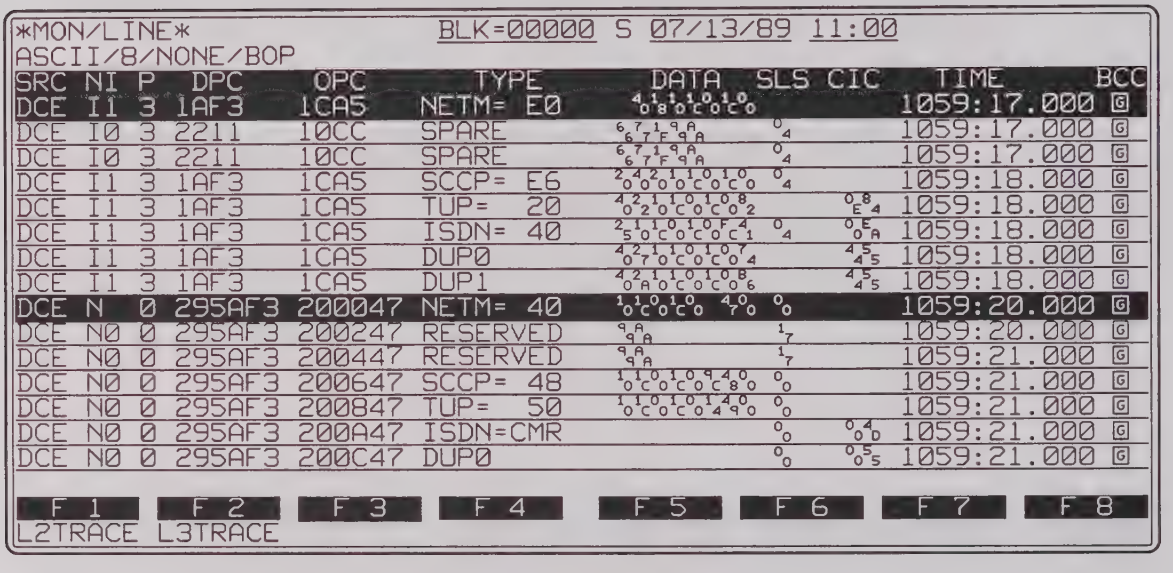


Figure 47-8 NETM Message Signal Units have been enhanced.

(B) Suppress

Individual MSUs that are suppressed in Layer 3 actions are deleted from the trace display. Figure 47-7 shows the softkey path to SUPPRES.

47.5 Structure of SS#7 Message Signal Units

Figure 47-9 shows the general structure of a Message Signal Unit. Figure 47-10 illustrates how CCITT International or National labels are entered and transmitted. MSU types are defined in Table 47-1. Table 47-2 through Table 47-5 define possible MSU headers for NETM, SCCP, ISUP, and TUP MSUs.

> **NOTE:** Structure, unless otherwise stated, reflects the MSU as displayed on the screen, not the actual transmission order of data.

Consult Figure 46-10 and Figure 46-11 for the format of Fill−In Signal Units (FISUs) and Link Status Signal Units (LSSUs). Any information pertinent to SS#7 Layer 2 is included in Section 46.

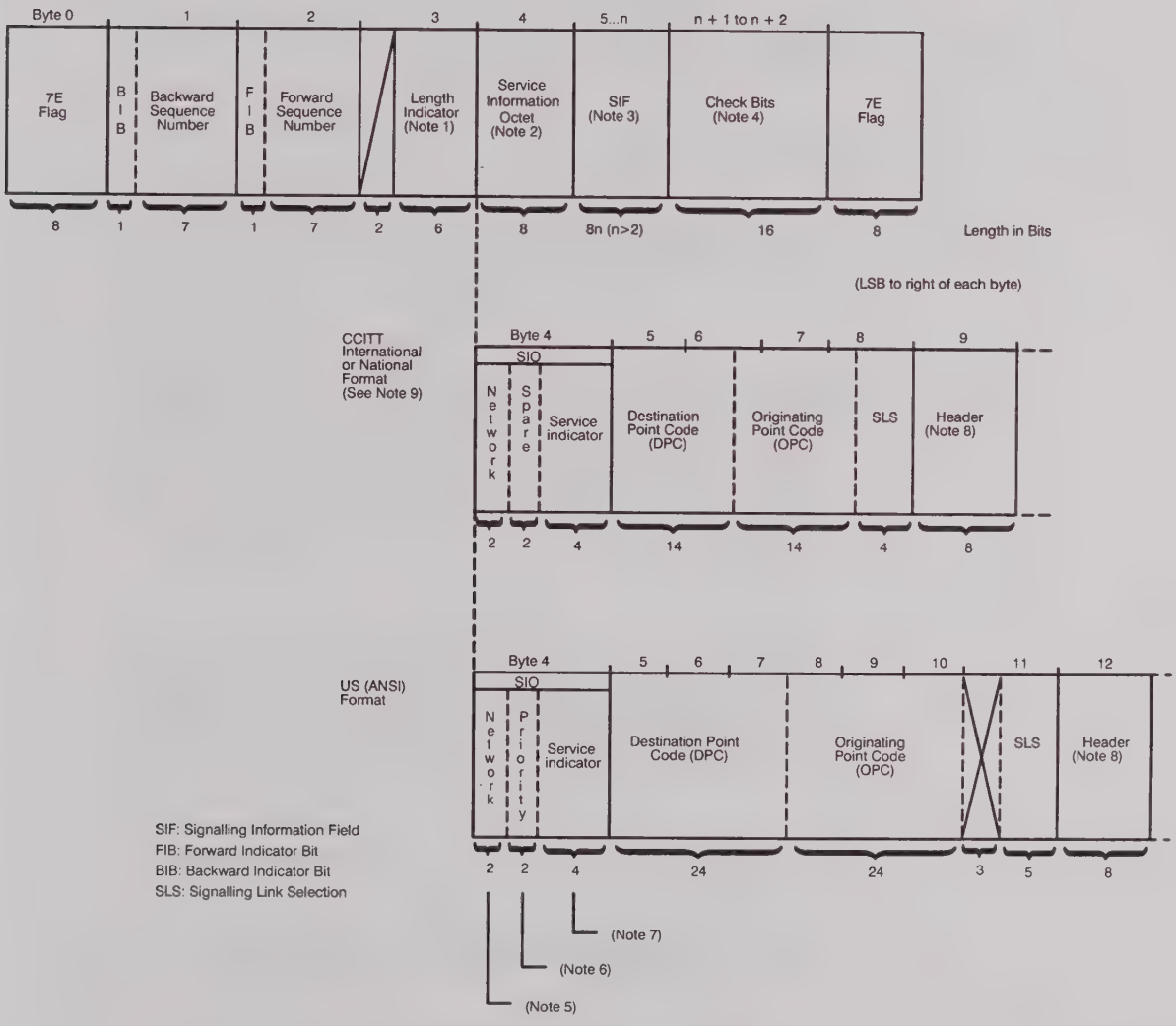


Figure 47-9 Format of a Message Signal Unit (MSU).

NOTES:

1. The Length Indicator for an MSU has a value of 3 to 63 (3 to 3F, hex).

2. The Service Information Octet contains a Service Indicator (Table 47-1), Bits 3−0 and a Network Indicator, Bits 7−6. In ANSI (US) format only, a Priority Code is present in Bits 5−4. Bits 5−4 are spare in CCITT International or National format.

3. The Signalling Information field contains a Label and a Header. There are two label formats: ANSI (US), a 56−bit format, and CCITT, a 32−bit format. The CCITT National label follows the same format as a CCITT International label (see expansions, this figure). The header is located at Octet 9 in CCITT International or National format and at Octet 12 in ANSI format.

4. The two octets preceding the closing flag are block check results. Only the first octet will appear in SS#7 line data. The second is replaced by a block check symbol (G, B, F). See Section 10 for an explanation of block checking in Bit−Oriented Protocols.

5. Network Indicators:

   00−01 = CCITT International format
   10−11 = ANSI (US Standard) or CCITT National format

6. The Priority Code has a value of 0−3 for ANSI format. These bits are undefined for CCITT International format. They are also undefined in CCITT National format.

7. A Service Indicator with Bits 3−2 = 01 (in either ANSI or CCITT International or National format) indicates a User Part. The subsequent structure of the signal unit is dependent on the User Part Type and is not reflected by this figure. Values for TUP and ISUP headers are given in Table 47-4 and Table 47-5.

8. The header gives the NETM Header (Table 47-2), SCCP Header (Table 47-3), or NTR/NTS Header.

   In NTR/NTS Messages (SIO = 0001), the four low−order bits determine the type:

   0001 = LTM (Signalling Link Test Message)

   0010 = LTA (Signalling Link Test Acknowledge)

   The four high−order bits are defined as 0001 for Test (NTR/NTS) Messages.

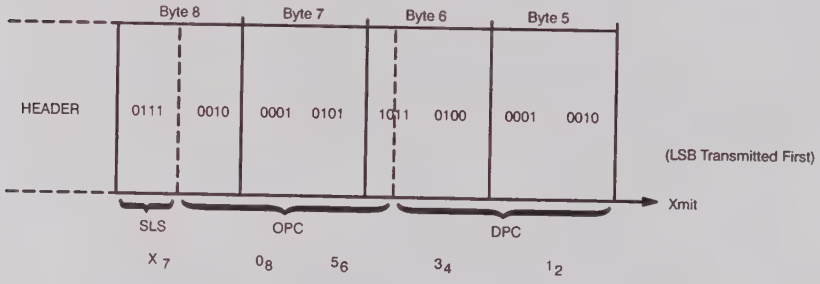
9. Since the label in CCITT International or National format is not byte−aligned, the values displayed in hexadecimal on the INTERVIEW screen are skewed. As a result, CCITT labels must be interpreted as shown in Figure 47-10.

Label values entered in spreadsheet conditions:

DPC= 3412

OPC= 0856

Bit sequence that will satisfy the DPC= and OPC= Conditions:



Data as it appears (in hex) on INTERVIEW screen or in send string:

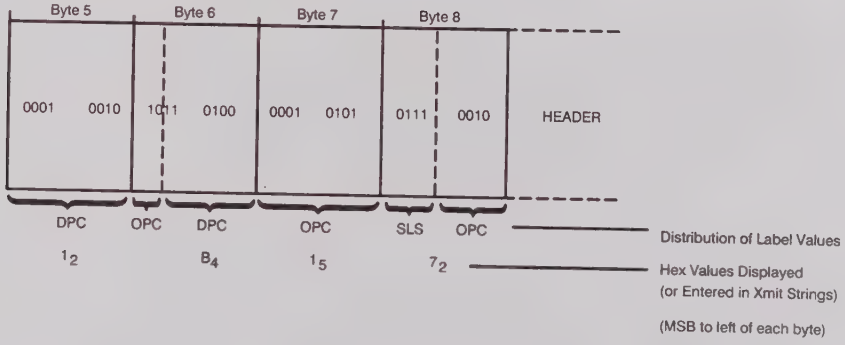


Figure 47-10 CCITT labels are entered and transmitted as shown.

Table 47-1
MSU Service Indicators
(Bits 3–0 of the SIO)

Mnemonic	Meaning	Binary Value
NETM (2)	Network Management	0000
NTR (2)	Network Test & Management Regular	0001
NTS (2)	Network Test & Management Special	0010
SCCP (2)	Signalling Connection Control Part	0011
TUP (1)	Telephone User Part	0100
ISUP (1) (2)	ISDN User Part	0101
DUP0	Data User Part (Call/Circuit)	0110
DUP1	Data User Part (Facility Regis/Canc)	0111
----	Spare	1000
----	Spare	1001
----	Spare	1010
----	Spare	1011
----	Spare	1100
----	Spare (3)	1101
----	Spare (4)	1110
----	Spare	1111

(1) Contains a CIC.
(2) Contains an SLS.
(3) ECIS6 for ANSI (US Standard) format.
(4) Reserved, if ANSI format.

Table 47-2
Network Management Headers
(Octet 9 for CCITT Format; Octet 12 for ANSI–US Format)

Mnemonic	Message	Hex	Added Bytes Shown	
			ANSI	CCITT
COO	Changeover Order	11	2	1
COA	Changeover Acknowledgment	21	2	1
CBD	Changeback Declaration	51	2	1
CBA	Changeback Acknowledgment	61	2	1
ECO	Emergency Changeover Order	12	1	0
ECA	Emergency Changeover Acknowledgment	22	1	0
RCT	Signalling Route–Set–Congestion–Test	13	0	0
TFC	Transfer Controlled	23	4	2
TFP	Transfer Prohibited	14	3	2
TCP (1)	Transfer Cluster Prohibited	24	3	–
TPA	Transfer–Prohibited Acknowledgment	24		
TFR	Transfer Restricted	34	3	2
TCR (1)	Transfer Cluster Restricted	44	3	–
TFA	Transfer Allowed	54	3	2
TCA (1)	Transfer Cluster Allowed	64	3	–
TAA	Transfer–Allowed Acknowledgment	64		
RSP (1) (4)	Route–Set–Test Destination Prohibited	15	3	–
RSR (1) (4)	Route–Set–Test Destination Restricted	25	3	–
RCP (1)	Route–Set–Test Cluster Prohibited	35	3	–
RCR (1)	Route–Set–Test Cluster Restricted	45	3	–
LIN	Link Inhibit	16	1	0
LUN	Link Uninhibit	26	1	0
LIA	Link Inhibit Acknowledgment	36	1	0
LUA	Link Uninhibit Acknowledgment	46	1	0
LID	Link Inhibit Denied	56	1	0
LFU	Link Forced Uninhibit	66	1	0
LLI (1)	Link Local Inhibit Test	76		–
LRI (1)	Link Remote Inhibit Test	86		–
DLC	Signalling–Data–Link Connection Order	18	2	2
CSS	Connection Successful	28	1	0
CNS	Connection Not Successful	38	1	0
CNP	Connection Not Possible	48	1	0

(1) ANSI (US) format only.
(2) CCITT format only.
(3) CCITT format only, National option.
(4) CCITT format uses the generic Route–Set–Test (RST) for both hex 15 and hex 25. This mnemonic is not presented on the softkey rack: it has been replaced by the ANSI designations RSP and RSR.

Table 47-3
SCCP Message Headers
(Octet 9 for CCITT International or National Format; Octet 12 for ANSI–US Format)

Mnemonic	Message	Hex	Added Bytes Shown	
			US	Int'l
CR	Connection Request	01	4+	4+
CC	Connection Confirm	02	7+	7+
CREF	Connection Refused	03	4+	4+
RLSD	Released	04	7+	7+
RLC	Release Complete	05	6	6
DT1	Data Form 1	06	6+	4+
DT2	Data Form 2	07	3+	5+
AK	Data Acknowledgment	08	4	5
UDT	Unitdata	09	1+	1+
UDTS	Unitdata Service	0A	1+	1+
ED	Expedited Data	0B	4+	3+
EA	Expedited Data Acknowledgment	0C	3	3
RSR	Reset Request	0D	7+	7+
RSC	Reset Confirmation	0E	6	6
ERR	Error	0F	4+	4+
IT	Inactivity Test	10	3	3

Table 47-4
Telephone User Part (TUP) Message Headers

Mnemonic	Message	Hex	Added Bytes Shown
IAM	Initial Address Message	11	3+
IAI	Initial Address Message With Additional Information	21	3+
SAM	Subsequent Address Message	31	1+
SAO	Subsequent Address Message With One Signal	41	1
GSM	General Forward Set–up Information Message	12	2+
COT	Continuity Signal	32	0
CCF	Continuity–Failure Signal	42	0
GRQ	General Request Message	13	1
ACM	Address Complete Message	14	1
CHG	Charging Message	24	0+
SEC	Switching–Equipment–Congestion Signal	15	0
CGC	Circuit–Group–Congestion Signal	25	0
NNC	National–Network–Congestion Signal	35	0
ADI	Address Incomplete Signal	45	0
CFL	Call–Failure Signal	55	0
SSB	Subscriber–Busy Signal	65	0
UNN	Unallocated–Number Signal	75	0
LOS	Line–Out–Of–Service Signal	85	0
SST	Send–Special–Information Tone Signal	95	0
ACB	Access Barred Signal	A5	0
DPN	Digital Path Not Provided Signal	B5	0
MPR	Misdialed Trunk Prefix	C5	0
EUM	Extended Unsuccessful Backward Set–up Info Message	F5	3
ANU	Answer Signal, Unqualified	06	0
ANC	Answer Signal, Charge	16	0
ANN	Answer Signal, No Charge	26	0
CBK	Clear–Back Signal	36	0
CLF	Clear–Forward Signal	46	0
RAN	Reanswer Signal	56	0
FOT	Forward–Transfer Signal	66	0
CCL	Calling Party Clear Signal	76	0
EAM	Extended Answer Message Indication	F6	0
RLG	Release–Guard Signal	17	0
BLO	Blocking Signal	27	0
BLA	Blocking–Acknowledgment Signal	37	0
UBL	Unblocking Signal	47	0
UBA	Unblocking–Acknowledgment Signal	57	0
CCR	Continuity–Check–Request Signal	67	0
RSC	Reset–Circuit Signal	77	0
MGB	Maintenance Group Blocking	18	1+
MBA	Maintenance Group Blocking Acknowledgment	28	1+
MGU	Maintenance Group Unblocking	38	1+
MUA	Maintenance Group Unblocking Acknowledgment	48	1+

Table 47-4 (Continued)
Telephone User Part (TUP) Message Headers

Mnemonic	Message	Hex	Added Bytes Shown
HGB	Hardware Failure Group Blocking	58	1+
HBA	Hardware Failure Group Blocking Acknowledgment	68	1+
HGU	Hardware Failure Group Unblocking	78	1+
HUA	Hardware Failure Group Unblocking Acknowledgment	88	1+
GRS	Circuit Group Reset Message	98	1+
GRA	Circuit Group Reset–acknowledgment Message	A8	1+
SGB †	Software Generated Group Blocking Message	B8	1+
SBA †	Software Generated Group Blocking–Acknowledgment	C8	1+
SGU †	Software Generated Group Unblocking Message	D8	1+
SUA †	Software Generated Group Unblocking–Acknowledgment	E8	1+
CFM	CCBS Facility Message	19	1+
CPM	Called Party Free Message	29	0+
CPA	Called Party Answer	39	0+
CSV	Closed User Group Selection/Validation Request	49	0+
CVM	Closed User Group Validation Check	59	3+
CRM	Closed User Group Selection/Validation Response	69	5
CLI	Connected Line Identity	79	1+

† National option.

Table 47-5
ISUP Message Headers

Mnemonic	Message	Hex	Added Bytes Shown
IAM	Initial Address	01	5+
SAM	Subsequent Address	02	0+
INR	Information Request	03	1+
INF	Information	04	1+
COT	Continuity	05	1
ACM	Address Complete	06	2+
FOT	Forward Transfer	08	0+
ANM	Answer	09	2+
UBM	Unsuccessful Backward Set–up Information	0A	1+
REL	Release	0B	0+
PAU	Pause	0D	0+
RES	Resume	0E	0+
RLSD	Released	0F	0
RLC	Release Complete	10	0
CCR	Continuity Check Request	11	0
RSC	Reset Circuit	12	0
BLO	Blocking	13	0
UBL	Unblocking	14	0
BLA	Blocking Acknowledgment	15	0
UBA	Unblocking Acknowledgment	16	0
GRS	Reset Circuit Group	17	0+
CGB	Circuit Group Blocking	18	1+
CGU	Circuit Group Unblocking	19	1+
CGBA	Circuit Group Blocking Acknowledgment	1A	1+
CGUA	Circuit Group Unblocking Acknowledgment	1B	1+
CMR	Call Modification Request	1C	1+
CMC	Call Modification Completed	1D	1+
RCM	Reject Connect Modify	1E	1+
FAR	Facility Request	1F	1+
FAA	Facility Accepted	20	1+
FRJ	Facility Reject	21	2+
FAD	Facility Deactivated	22	1+
FAI	Facility Information	23	2+
CSVR	Closed User Group Selection/Validation Request	25	0+
CSVS	Closed User Group Selection/Validation Response	26	1+
DRS	Delayed Release	27	0+
PAM	Pass Along	28	0+
GRA	Reset Circuit Group Acknowledgment	29	0+

48 SMDS

INTERVIEW 8000 Series units now support applications for monitoring and emulation on the SMDS (Switched Multimegabit Data Service) SNI (Subscriber Network Interface) at T1 data rates and DXI (Data Exchange Interface) at T1 and E1 data rates with no additional hardware requirement.

- The SMDS SNI for T1 is defined in Bellcore document TR−TSV−000772 as a public service offering. The SNI is based on 53-byte cells and uses a PLCP (Physical Layer Convergence Protocol) to map cells onto the T1 framed data streams.

  NOTE: The SMDS specifications require all octets are transmitted *most significant bit* first.

  To record at these rates with SCSI drive, select the following:

  SETUP: LINE:
  Format: SYNC
  Sync Char: $^{6}_{F}$ $^{1}_{4}$
  HS Frame Mode: SMDS
  RECORD:
  Capture Memory: DISK
  Disk No: HRD
  Data To Record: BIT IMAGE
  DAT Record Size: 64K
  Stop at: user choice

- The DXI protocol is used between routers and SMDS DSUs. It is in BOP format with an optional 32−bit CRC.

  To record at these rates with SCSI drive, select the following:

  SETUP: LINE:
  Format: BOP
  High Speed Frame Mode: YES
  RECORD:
  Capture Memory: DISK
  Disk No: HRD
  Data To Record: BIT IMAGE
  DAT Record Size: 64K
  Stop at: user choice
  BCC:
  CCITT Polynomial: CCITT−16 or CCITT−32

Monitoring and emulating at up to 2.048 Mbps circuit may be accomplished using application programs specifically designed for the INTERVIEW 8000 units.

49 ATM

INTERVIEW 8800 *TURBO* units now support applications for monitoring and emulation in ATM (Asynchronous Transfer Mode) with no additional hardware required:
Trace and Statistics: Asynchronous Transfer Mode (T1/E1) [OPT−951−232−1] and
Emulation with Trace and Statistics: Asynchronous Transfer Mode (T1/E1) [OPT−951−233−1].

ATM for T1 is defined in the ATM Forum DS1 ATM UNI (draft) Specification and the ITU Draft Documentation I.432 and G.804. ATM is based on 53-byte cells; the INTERVIEW decodes it using either a PLCP (Physical Layer Convergence Protocol) to map cells onto the T1 framed data streams (via 576−byte superframe) or the standard ATM HEC (Header Error Control) framing.

49.1 Recording ATM over T1/E1 Data with the INTERVIEW

To record T1/E1 data at ATM rates with a SCSI drive in an INTERVIEW 8800 *TURBO* unit, setup the following parameters:

SETUP: LINE:

Format: SYNC

Sync Character: $^{6}F\ ^{1}4$ (when using PLCP framing; otherwise not applicable)

HS Frame Mode: ATM

HEC CRC−8 Coset: $^{5}5$

Payload Scrambled: user choice

PLCP Framing: user choice

RECORD:

Capture Memory: DISK

Disk No: HRD

Data To Record: BIT IMAGE

DAT Record Size: 64K

Stop at: user choice

49.2 Overview

Asynchronous Transfer Mode (ATM) is a technology that offers the "bandwidth on demand" features of packet−switching with the high speeds required for LAN and WAN networks today. This cell−relay technology operates independently of the type of

transmission being generated at the upper level *and* of the type and speed of the physical layer medium being used. This allows sending of virtually any type of transmission (*e.g.*, voice, data, video) in a single integrated data stream operating over any medium ranging from existing T1/E1 lines to SONET OC–3 at speeds of 1.5 to 155 Mbps — and faster.

The technology permits both public (*i.e.*, RBOC or local carrier) and private (*i.e.*, LAN or LAN–to–internal switch) ATM networks. This capability gives a seamless and transparent (to the user) connection from one end user to another, whether in the same building or across two continents.

(A) OSI Layer Similarity

At the end user site, ATM operates with a layered structure similar to the OSI 7–layered model. However, ATM only addresses the lower two layers of this model:

- Layer 1 — Physical Medium
- Layer 2 — Data Link

All other layers are only part of the encapsulated information portion of the cell, which is passed transparently through the ATM network.

These two OSI layers are handled by three layers for ATM:

- Physical Layer — This layer defines the medium for transmission, any medium–dependent parameters (*e.g.*, the Quality of Service requirements), and the framing used to find the data contained within the medium.
- ATM Layer — This layer provides the basic 53–byte cell format and defines the cell header content.
- ATM Adaptation Layer (AAL) — This layer adapts the higher–level data into formats compatible with the ATM Layer requirements. It is dependent on the type of service(s) being transported by the higher layer.

When the end user sends traffic over the ATM network, the higher–level data unit is passed down to the AAL Layer, which prepares the data for the ATM Layer as appropriate for the AAL protocol being used. This can include padding the data unit to a fixed length, adding headers and/or trailers for error checking and higher–level routing, and segmentation (with or without additional information added to the subunits).

The prepared data unit or segment is then passed down to the ATM Layer, which affixes a necessary 5–byte ATM header to the segment. This 53–byte ATM cell is then passed down to the Physical Layer for transmission (HEC framing) or packaging into a Physical Layer Convergence Protocol superframe (PLCP framing).

(B) AAL Protocols

Four AAL protocols have been defined for use in ATM networks. These protocols loosely correspond to specific data classes, but are not necessarily exclusive to that data type:

- AAL 1 — Constant bit rate, connection–oriented, synchronous traffic (*e.g.*, uncompressed voice)
- AAL 2 — Variable bit rate, connection–oriented, synchronous traffic (*e.g.*, compressed video)
- AAL 3/4 — Variable bit rate, connection–oriented, asynchronous traffic (*e.g.*, X.25 data) or connectionless packet data (*e.g.*, LAN traffic) with additional information on segmentation and cell order
- AAL 5 — Variable bit rate, connection–oriented, asynchronous traffic (*e.g.*, X.25 data) or connectionless packet data (*e.g.*, LAN traffic) with a simplified information scheme for resegmentation

Examples of data flow for the commonly used AAL 3/4 and AAL 5 protocols are given on the next few pages.

49.3 AAL 1 Protocol

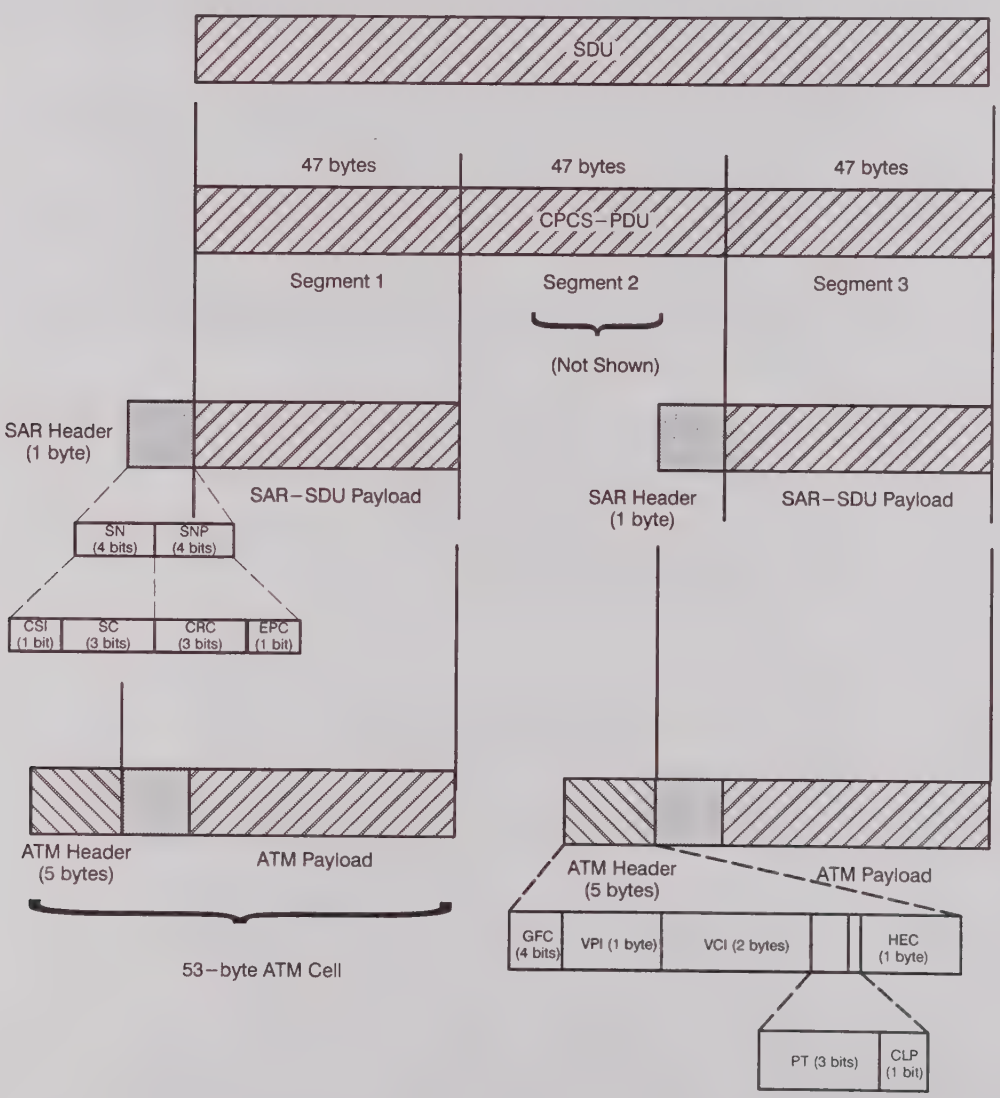


Figure 49-1 AAL 1 protocol: flow of a 141–byte SDU from the end user to the ATM layer.

The AAL 1 example shown in Figure 49-1 follows the flow of a 141–byte message (Service Data Unit) from the end user to the ATM Layer. The SDU is passed unchanged through the Convergence Sublayer to the Segmentation and Reassembly Sublayer (SAR). The steps to accomplish this are documented in the following pages in Figure 49-2 through Figure 49-4.

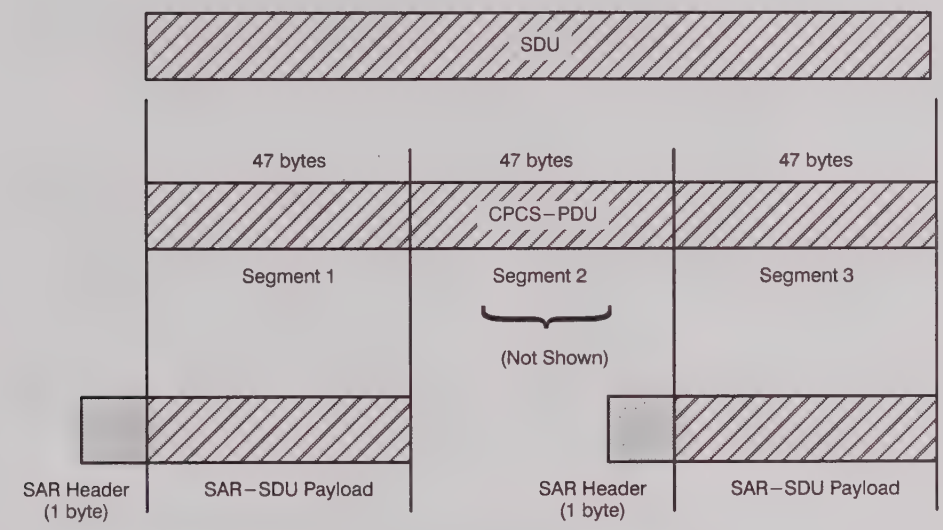


Figure 49-2 AAL 1 protocol: the SAR sublayer divides the entire PDU into 47–byte segments from the beginning and adds a 1–byte header to each segment.

The AAL 1 example follows the flow of a 141–byte message (Service Data Unit) from the end user to the ATM Layer. The SDU is passed unchanged through the Convergence Sublayer to the Segmentation and Reassembly Sublayer (SAR).

Figure 49-2 shows that the SAR sublayer divides the entire PDU into 47–byte segments from the beginning and adds a 1–byte header to each segment. This header adds information concerning the sequencing of the segments and providing an error–check mechanism.

The 1–byte SAR header consists of the following, as shown in Figure 49-3:

- SN — Sequence Number, a 4–bit field which indicates the sequence number information for this segment.
- SNP — Sequence Number Protection, a 4–bit field which provides the error–checking mechanism for the SAR header.

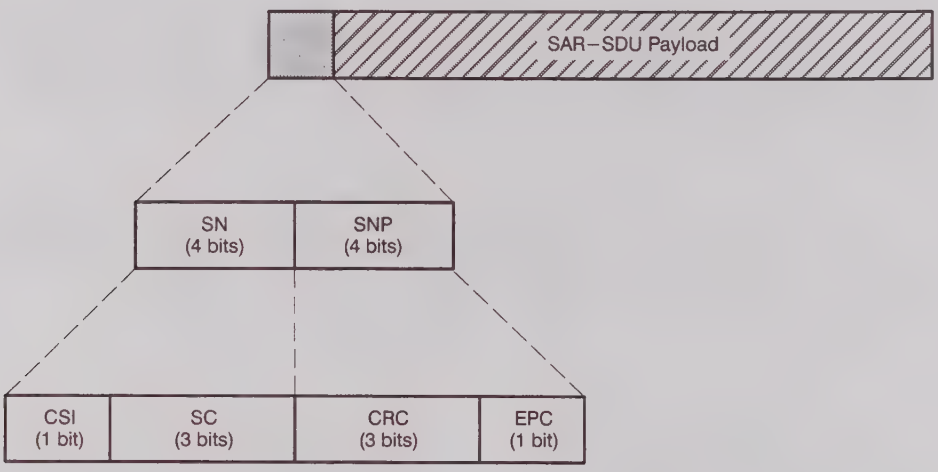


Figure 49-3 AAL 1 protocol: the 1–byte SAR header consists of two 4–bit fields: SN and SNP. Each of these fields is in turn subdivided into a 3–bit and a 1–bit field.

The Sequence Number portion of the header is further subdivided into two subfields, as shown in Figure 49-3:

- CSI — Convergence Sublayer Indications, a 1–bit subfield which is used to convey CS–specific information. This is not utilized for all AAL 1 implementations.
- SC — Sequence Count, a 3–bit sequence number for the entire CS–PDU. This is generated by the CS and remains constant for all segments created from that CS–PDU.

The Sequence Number Protection portion of the header is also subdivided into two subfields, as shown in Figure 49-3:

- CRC — Cyclical Redundancy Check, a 3–bit sequence that functions as an error check for the SN field *only*.
- EPC — Even Parity Check, a 1–bit check of the previous 7 bits of the header, *i.e.*, the SN field and the CRC subfield.

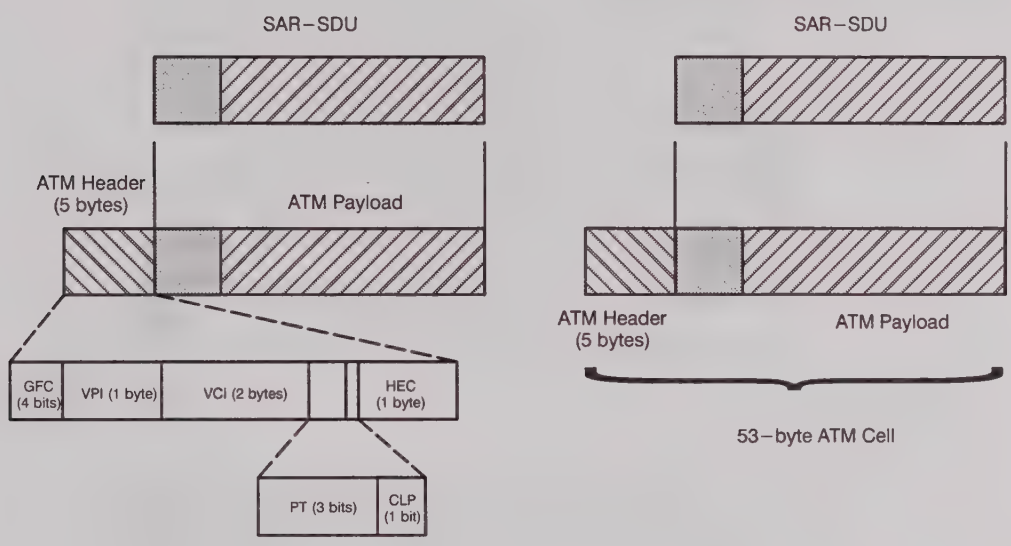


Figure 49-4 AAL 1 protocol: a 5–byte ATM header is added to the SAR–PDU segment (which becomes the transparent 48–byte payload for the ATM cell) to create the ATM cell.

This SAR–PDU (segment) becomes the transparent 48–byte payload for the ATM cell. As shown in Figure 49-4, a 5–byte ATM header is then added to the segment to create the ATM cell. This header contains the information necessary to transport the cell to its destination, identify the payload type, assign the cell a loss priority, and provide an error–checking mechanism.

The ATM header consists of the following fields:

- GFC — Generic Flow Control, of local significance only and used to provide standardized local functions on the end user premises.
- VPI — Virtual Path Identifier, which with the VCI provides the routing from node to node throughout the network from end user to destination.
- VCI — Virtual Channel Identifier (see above).
- PT — Payload Type, used to indicate if the cell contains user information or flow control information. This 3–bit field may also be used to indicate network congestion or network resource management information.
- CLP — Cell Loss Priority, to optionally indicate cells that may be discarded when congestion occurs.
- HEC — Header Error Control, used to detect bit errors in the ATM Header only.

49.4 AAL 3/4 Protocol

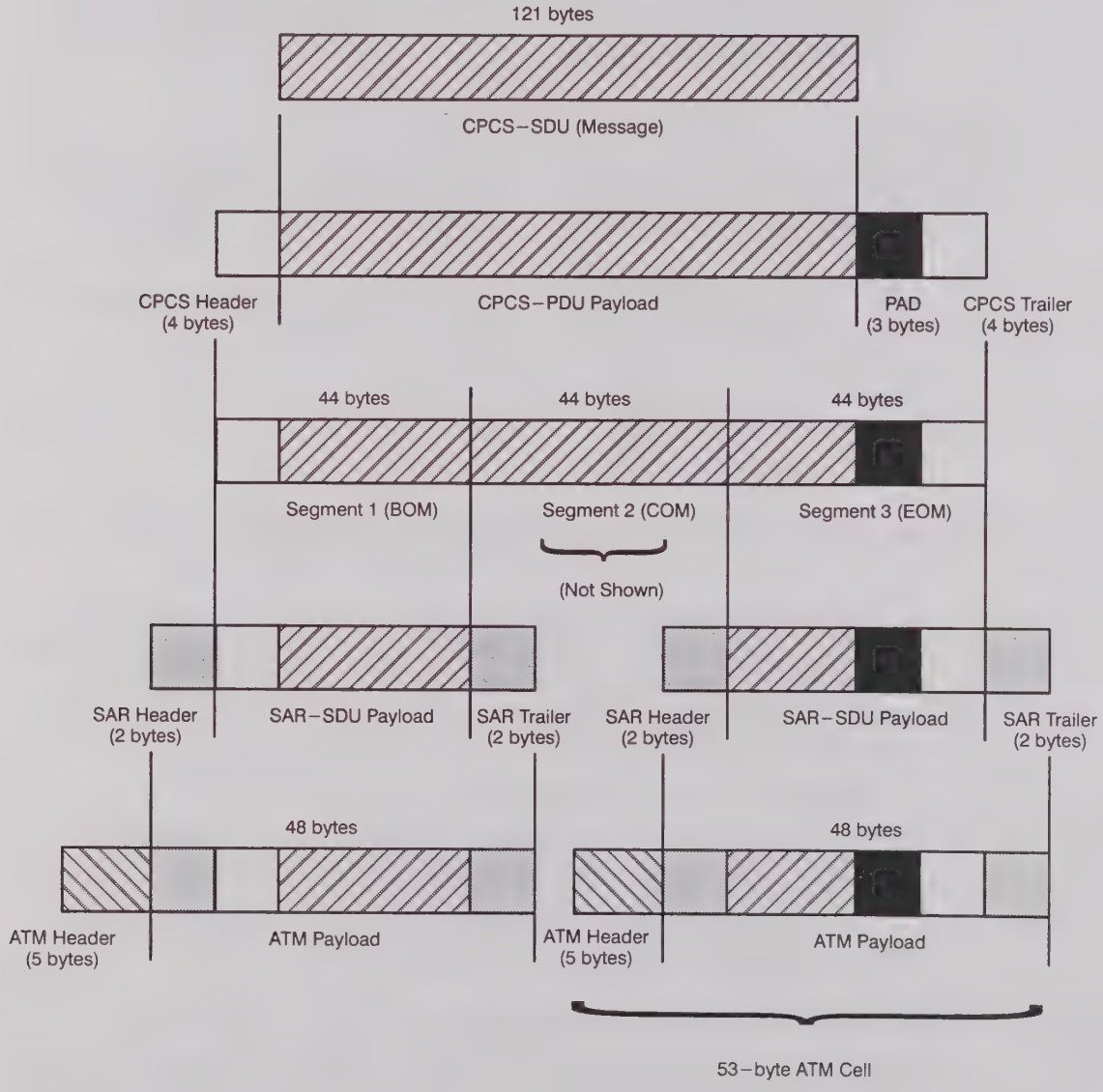


Figure 49-5 AAL3/4 protocol: transformation of a 121–byte CPCS–SDU message from the end user into a 53–byte ATM cell.

The AAL 3/4 example shown in Figure 49-5 follows the flow of a 121–byte message (Service Data Unit) from the end user to the ATM Layer. The steps to accomplish this are documented in the following pages in Figure 49-6 through Figure 49-9.

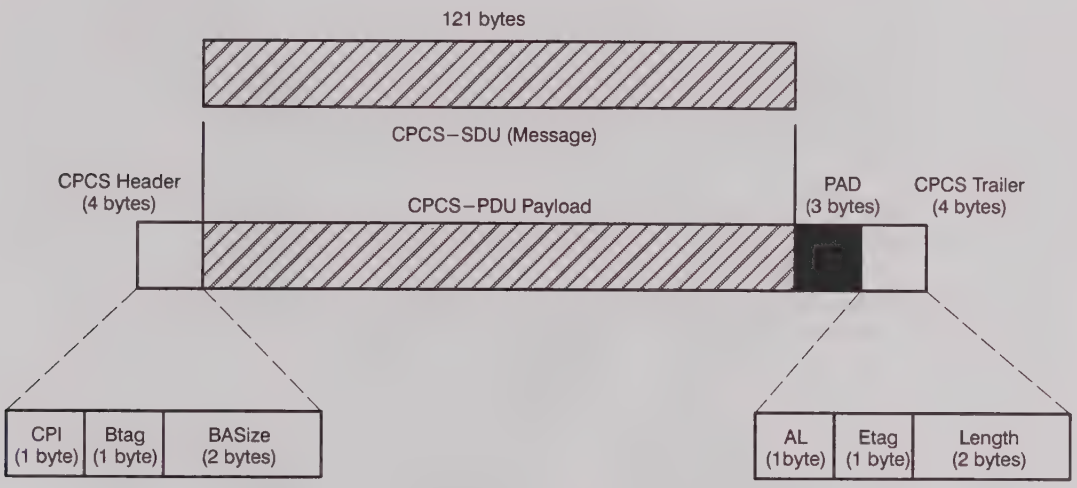


Figure 49-6 AAL3/4 protocol: The SDU is passed to the Convergence Sublayer, where a 4–byte header and a 4–byte trailer are added to the front and end of the SDU.

The SDU is passed to the Convergence Sublayer, where a 4–byte header and a 4–byte trailer are added to the front and end of the SDU (see Figure 49-6). In addition, the data segment is padded with 0 to 3 bytes of data at the end, as necessary, to make the entire Protocol Data Unit (PDU) size a multiple of 4 bytes. (In our example, the pad is 3 bytes.)

The Common Part Convergence Sublayer (CPCS) header consists of the following:

- CPI — Common Part Indicator, used to interpret the remainder of the fields in the header and the trailer added for this sublayer.
- Btag — Beginning Tag, an "error check" for this segment. The value in this field is also placed in the Etag field of the trailer, allowing a quick comparison after receipt to determine if the PDU has been corrupted.
- BASize — Buffer Allocation Size, encoded to indicate the CPCS–PDU payload length.

The CPCS trailer consists of the following:

- AL — Alignment, used to make the trailer size 4 bytes and passed transparently through the network.
- Etag — End Tag (see Btag)
- Length — Used to indicate the length of the PDU payload. This field is encoded to indicate the number of counting units in the length of the payload, with the counting unit size indicated in the CPI of the header.

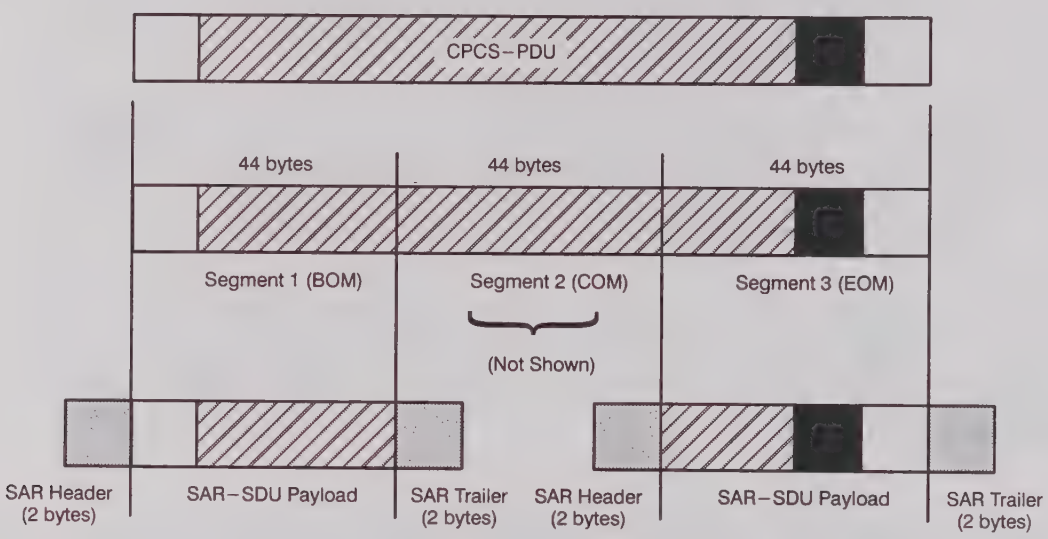


Figure 49-7 AAL3/4 protocol: The entire CPCS–PDU is then passed to the Segmentation and Reassembly (SAR) sublayer for further processing.

Figure 49-7 shows that the entire CPCS–PDU is then passed to the Segmentation and Reassembly (SAR) sublayer for further processing. This sublayer divides the entire PDU into 44–byte segments from the beginning and adds a 2–byte header and a 2–byte trailer to each segment. These headers and trailers add information concerning the segment order and provide a cyclical redundancy check (CRC) for the segment.

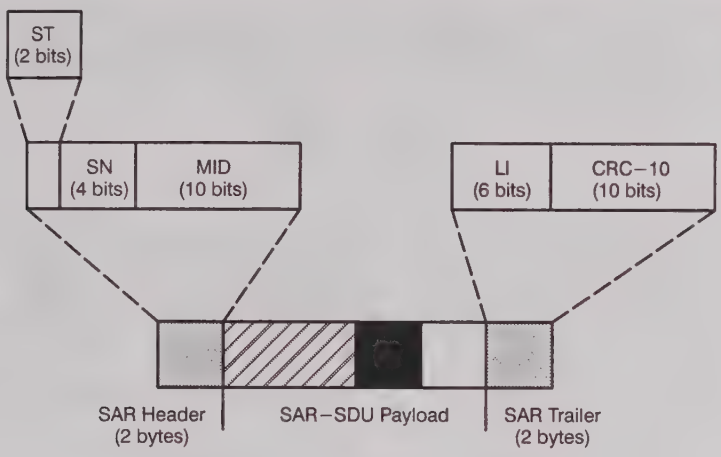


Figure 49-8 AAL3/4 protocol: SAR header.

The SAR header consists of the following:

- ST — Segment Type, one of the four possible segment types: Beginning of Message (BOM), Continuation of Message (COM), End of Message (EOM), or Single Segment Message (SSM).

  NOTE: In the example above, the EOM is a full 44 bytes in length. This will not necessarily be true of all PDUs, as the only requirement is that of maintaining a PDU length divisible by 4 bytes. Therefore, this segment will contain whatever portion remains after all of the previous 44–byte segments have been created by the segmentation process padded with zeroes as necessary.

- SN — Sequence Number, allowing the stream of SAR Service Data Units (SDUs) to be numbered using modulo 16 in order to provide a "loss of segment" check for each full PDU that is segmented.

- MID — Multiplexing Identification, used to multiplex CPCS connections on a single ATM Layer connection, when applicable

The SAR trailer consists of two parts:

- LI — Length Indication, binary encoded to indicate the number of bytes of the CPCS–PDU are contained in the payload portion of the segment. For the BOM and COM segments this value must be 44. For EOM segments, the value can range from 4 to 44 as appropriate. For SSM segments, permissible values range from 8 to 44.

- CRC — Cyclical Redundancy Check, a 10–bit sequence that functions as an error check for the entire SAR–SDU, including the header, payload, and the LI field of the trailer.

This SAR–PDU is then handed down to the ATM Layer for further processing.

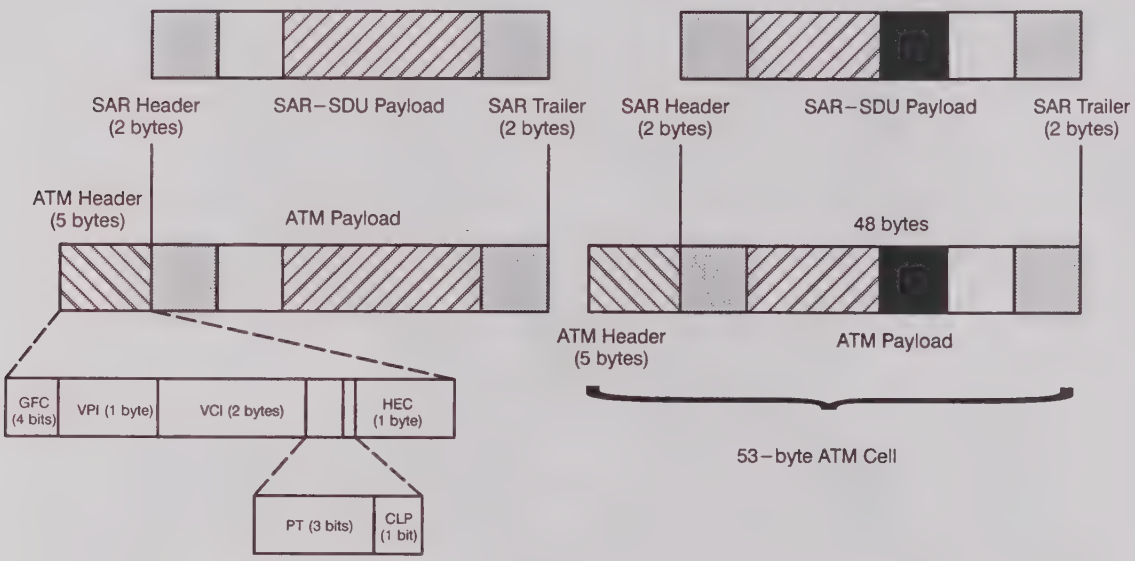


Figure 49-9 AAL3/4 protocol: The SAR–PDU (segment) becomes the transparent 48–byte payload for the ATM cell.

This SAR–PDU (segment) becomes the transparent 48–byte payload for the ATM cell (see Figure 49-9). A 5–byte ATM header is then added to the segment to create the ATM cell. This header contains the information necessary to transport the cell to its destination, identify the payload type, assign the cell a loss priority, and provide an error–checking mechanism.

The ATM header consists of the following fields:

- GFC — Generic Flow Control, of local significance only and used to provide standardized local functions on the end user premises.
- VPI — Virtual Path Identifier, which with the VCI provides the routing from node to node throughout the network from end user to destination. See Table 49-1 for pre–defined header field values.
- VCI — Virtual Channel Identifier (see VPI, above).
- PT — Payload Type, used to indicate if the cell contains user information or flow control information. This 3–bit field may also be used to indicate network congestion or network resource management information. See Table 49-1 for pre–defined header field values and Table 49-2 for PT encoding.
- CLP — Cell Loss Priority, to optionally indicate cells that may be discarded when congestion occurs. See Table 49-1 for pre–defined header field values.
- HEC — Header Error Control, used to detect bit errors in the ATM Header only.

Table 49-1
Pre–defined Header Field Values †

Use	Value [1,2,3,4]			
	Octet 1	Octet 2	Octet 3	Octet 4
Unassigned cell indication	00000000	00000000	00000000	0000xxx0
Meta–signalling (default) [5,7]	00000000	00000000	00000000	00010a0c
Meta–signalling [6,7]	0000yyyy	yyyy0000	00000000	00010a0c
General Broadcast signalling (default) [5]	00000000	00000000	00000000	00100aac
General Broadcast signalling [6]	0000yyyy	yyyy0000	00000000	00100aac
Point–to–point signalling (default) [5]	00000000	00000000	00000000	01010aac
Point–to–point signalling [6]	0000yyyy	yyyy0000	00000000	01010aac
Invalid Pattern	xxxx0000	00000000	00000000	0000xxx1
Segment OAM F4 flow cell [7]	0000aaaa	aaaa0000	00000000	00110a0a
End–to–End OAM F4 flow cell [7]	0000aaaa	aaaa0000	00000000	01000a0a

1 *a* indicates that the bit is available for use by the appropriate ATM layer function.
2 *x* indicates "don't care" bits.
3 *y* indicats any VPI value other than 00000000.
4 *c* indicates that the originating signalling entity shall set the CLP bit to 0. The network may change the value of the CLP bit.
5 Reserved for user signalling with the local exchange.
6 Reserved for signalling with other signalling entities (e.g., other users or remote networks).
7 The transmitting ATM entity shall set bit 2 of octet 4 to zero. The receiving ATM entity shall ignore bit 2 of octet 4.

Table 49-2
Payload Type Indicator Encoding †

PTI Coding (MSB first)	Interpretation
000	User data cell, congestion not experienced, SDU–type = 0
001 *	User data cell, congestion not experienced, SDU–type = 1
010	User data cell, congestion experienced, SDU–type = 0
011 *	User data cell, congestion experienced, SDU–type = 1
100	Segment OAM F5 flow related cell
101	End–to–end OAM F5 flow related cell
110	Reserved for future traffic control and resource management
111	Reserved for future functions

* This coding indicates this is the end of a SAR–PDU segment for AAL 5

† Table information from *ATM Forum UNI Specification*, Subsections 3.4.4 and 3.4.5, respectively

49.5 AAL 5 Protocol

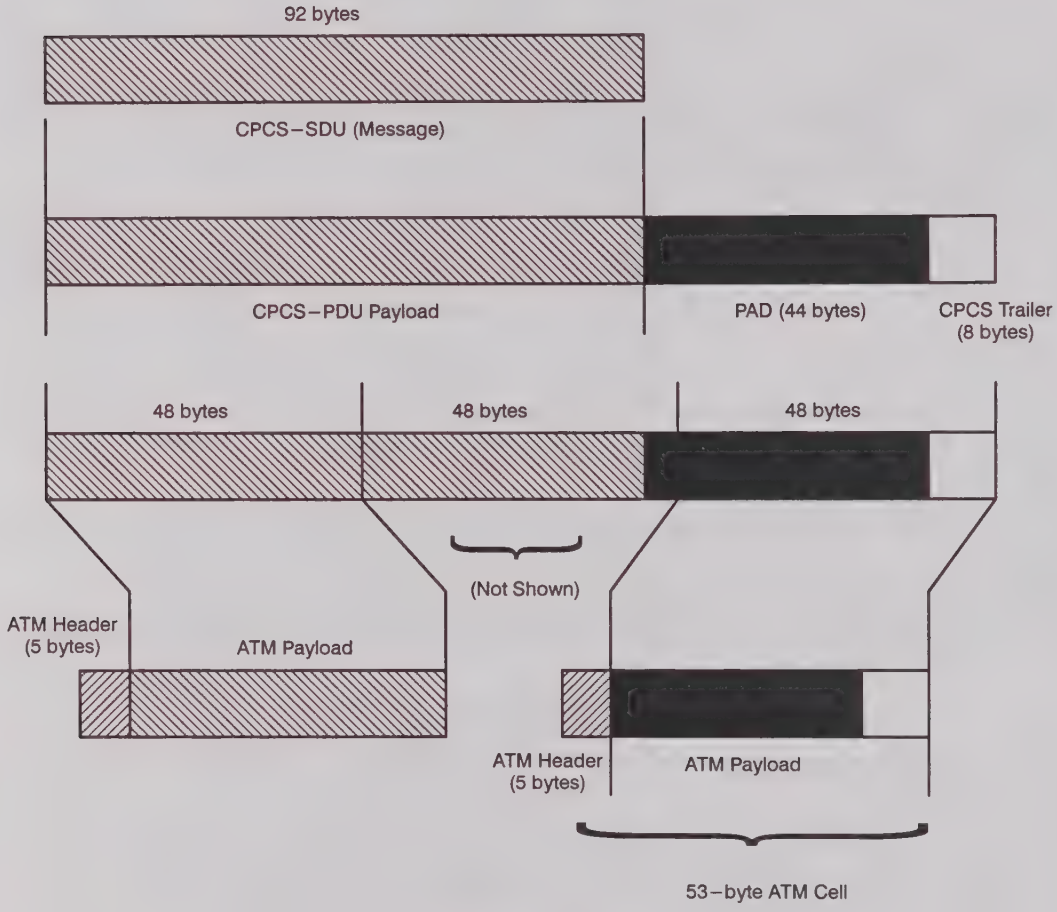


Figure 49-10 AAL 5 protocol

The AAL 5 example follows the flow of a 92–byte message (Service Data Unit) from the end user to the ATM Layer. The procedure is further broken down and explained in Figure 49-11 and Figure 49-12.

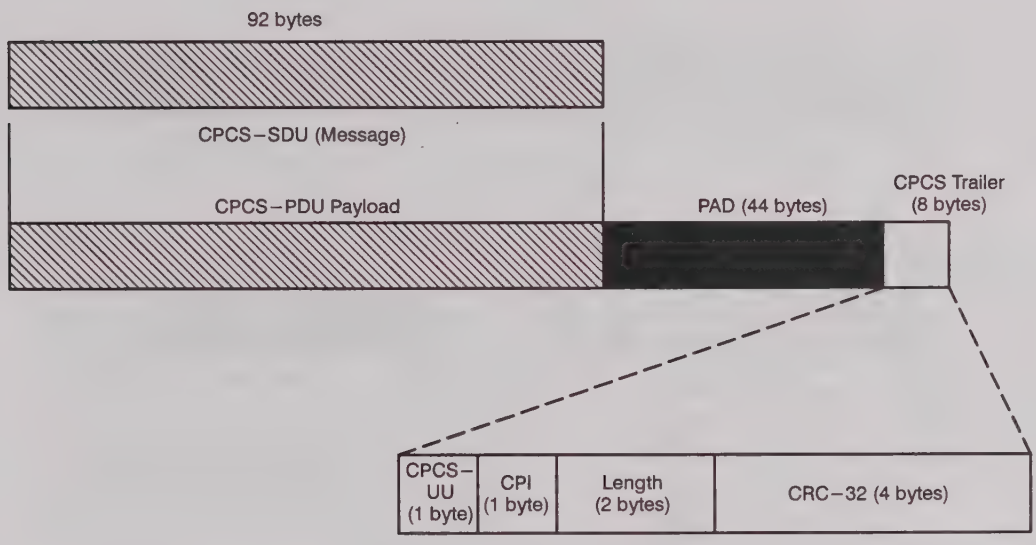


Figure 49-11 AAL 5 protocol: the SDU is passed to the Convergence Sublayer, where an 8–byte trailer is added to the end of the SDU.

Figure 49-11 illustrates the SDU is passed to the Convergence Sublayer, where an 8–byte trailer is added to the end of the SDU. In addition, the data segment is padded with 0–47 bytes of data at the end, as necessary, to place the trailer at the *end* of the last 48–byte segment. (In our example, the pad is 44 bytes.)

The CPCS trailer consists of the following:

- CPCS–UU — CPCS User–to–User indication, used to transparently transfer CPCS information from the origination user to the destination user.
- CPI — Common Part Indicator, used to align the CPCS–PDU trailer to the 32–bit boundary.
- Length — Used to indicate the length of the CPCS payload (not including the PAD bytes).
- CRC — Cyclical Redundancy Check, a 32–bit error check for the entire contents of the CPCS–PDU, including the payload, the PAD field, and the first 4 bytes of the trailer.

This entire CPCS–PDU is then handed to the SAR sublayer for segmentation.

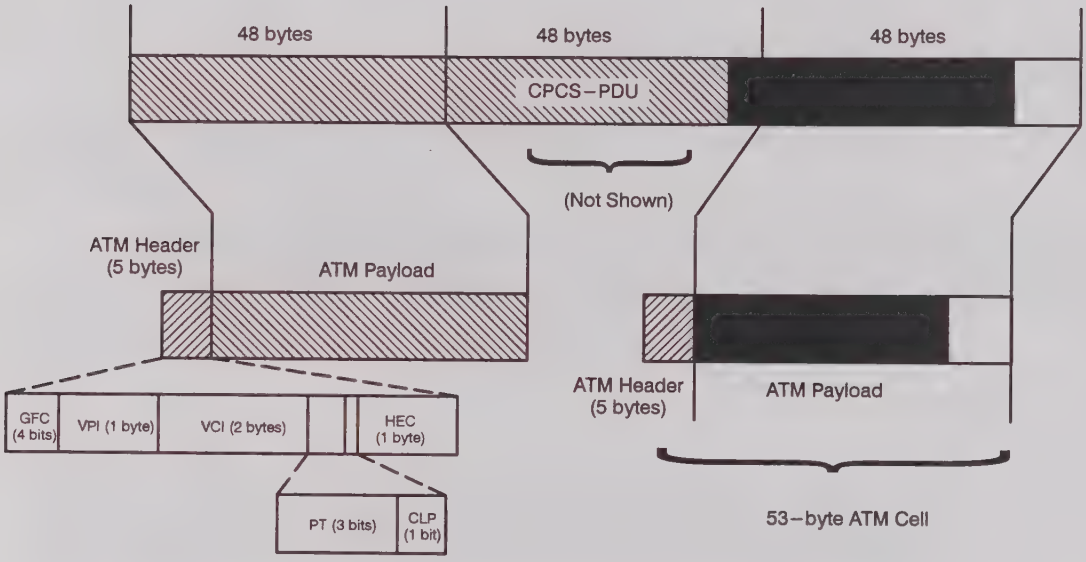


Figure 49-12 AAL 5 protocol: the ATM header is the same as that discussed for AAL 3/4 with one exception—the PT code can identify the end cell of a SAR–PDU segment. (See Table 49-2.)

The SAR sublayer segmentation for AAL 5 consists solely of dividing the CPCS–PDU into 48–byte segments. These segments are then handed down to the ATM Layer, where the ATM Header is added to create the 53–byte ATM cells.

NOTE: Refer to the AAL 3/4 explanation for details of the AAL 5 header format. The ATM Header structure is the same for all AAL types, with one exception for the AAL 5 format: two of the Payload Type codes identify the end cell of a SAR–PDU segment. The footnote of Table 49-2 for the Payload Type encoding describes the *end* cell PT codes as:

001 User data cell, congestion not experienced, SDU–type = 1
011 User data cell, congestion experienced, SDU–type = 1

49.6 Operations and Maintenance (OAM) Cells

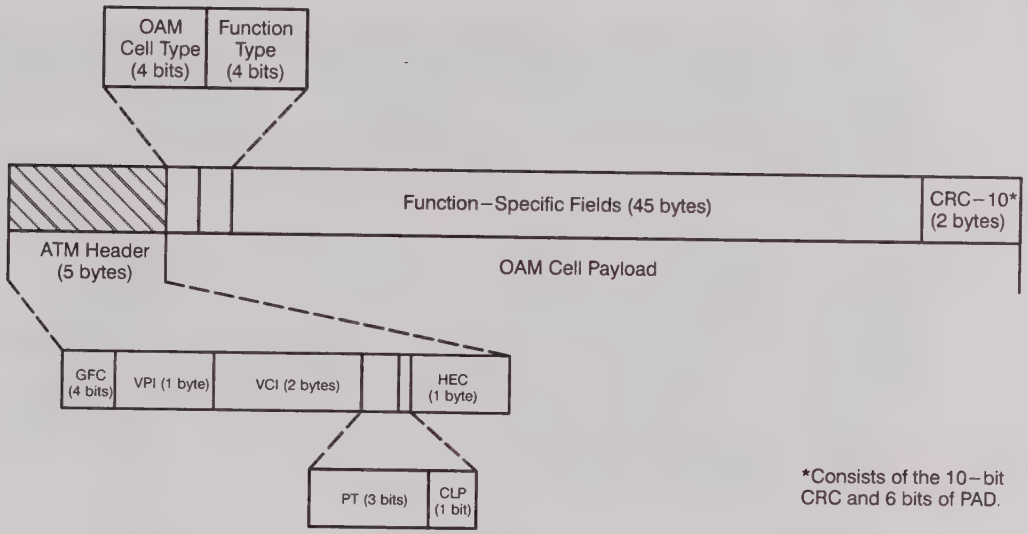


Figure 49-13 Operations and Maintenance (OAM) cells.

The INTERVIEW 8000 *TURBO* Series also supports one additional format in ATM over T1/E1. The Operations and Maintenance (OAM) cells are used to provide various maintenance functions within the ATM network, including connectivity verification and alarm surveillance. These cells consist of a single segment with the following 48–byte format:

- OAM Cell Type — A 4–bit indicator of the type of OAM cell (*e.g.*, 0001 to indicate Fault Management).
- Function Type — A 4–bit indicator of the purpose for this particular OAM cell (*i.e.*, Alarm Indication Signal, Far End Receive Failure, Loopback).
- Function–Specific Fields — Functions and information for this cell (up to 45 bytes), used to denote destination and failure information.
- CRC — Cyclical Redundancy Check, a 10–bit error check for the OAM payload, including the type indicators and the function–specific fields.

This 48–byte OAM cell is passed to the ATM Layer, which adds the ATM Header to create the 53–byte ATM cell.

49.7 ATM Framing: HEC and PLCP

Two types of framing for ATM are supported by the INTERVIEW 8000 *TURBO* Series: HEC (Header Error Control) framing and PLCP (Physical Layer Convergence Protocol) framing.

For HEC framing, the ATM cells (regardless of AAL format) are streamed across the physical medium of the ATM network as processed by the ATM Layer and the Header Error Control values are used to achieve synchronization. With this format, alignment is achieved when the appropriate number of successive cells with good HEC values are detected (six is recommended). Cell Delineation Loss or loss of synchronization occurs when an appropriate number of successive HEC errors are detected (seven is recommended).

Figure 49-14 illustrates PLCP, the alternate framing mode. PLCP framing bundles 10 ATM cells into a superframe that adds 4 bytes of information to the beginning of each cell. The first two bytes of this information (A1 and A2) act as a 2–byte synchronization sequence. The next byte (P9 to P0) is the path overhead indicator for each ATM cell in the superframe. The fourth byte (Z4–Z1, F1, B1, G1, M2–M1, and C1) is the path overhead for the superframe. These consist of Growth octets (Z4–Z1), Bit–Interleaved Parity (B1), PLCP Path Status (G1), Cycle/Stuff Counter (C1), and three fields (F1, M1, and M2) that are irrelevant to the ATM network.

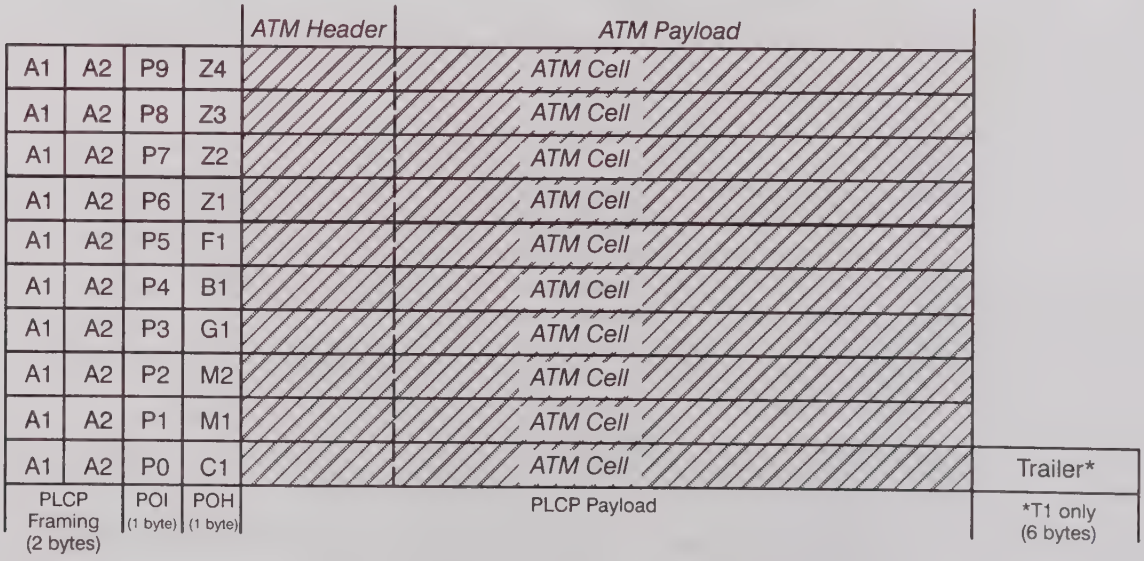


Figure 49-14 PLCP frame format.

In addition, a 6–byte trailer is added when T1 is utilized. This trailer consists of the same character (ᶜc) repeated six times and is used to increase the superframe to the required 576–byte size for T1. It serves no other purpose.

Index A:
Technical Manual Part I

Symbols

A abort, overlay for BOP abort, 6–13

B badbcc, overlay for bad BCC or FCS, 6–13

M bit mask symbol, in Receive string condition, 25–7

)) close double parens symbol, 3–5, 26–4, 29–5

*/ closing delimiter for comment, 28–14

X don't care symbol
in Receive string condition, 25–7
in spreadsheet string search, 37–35

F flag, 7E flag symbol, 6–14

G goodbcc, overlay for good BCC or FCS, 6–13

+ highlighted plus symbol, indicates wrap in logical line, 30–3

= not equal flag symbol
in Receive string condition, 25–7
in spreadsheet string search, for beginning of frame, 37–35

((open double parens symbol, 3–5, 26–4, 29–5

/* opening delimiter for comment, 28–14

▨ pad, in Outsync Char field, 5–13

? key, Easy View, 4–7

/ (root) directory, 15–4

S sync, sync symbol, 5–8, 6–14

~ tilde symbol, 3–5, 28–3

♫ time–fill, 6–8

A

AAL
in ATM, 49–2
protocols in ATM, 49–3

AAL 1, protocol in ATM, 49–3, 49–4–49–8

AAL 2, protocol in ATM, 49–3

AAL 3/4, protocol in ATM, 49–3, 49–8–49–12

AAL 5, protocol in ATM, 49–3, 49–14–49–16

Abort
adjunct to monitor–frame condition
LAPD, 43–14, 43–15
SDLC, 39–15, 39–17
X.25 Layer 2, 37–14, 37–15
adjunct to send–frame action
LAPD, 43–25
SDLC, 39–28
X.25 Layer 2, 37–28
adjunct to send–string action, Layer 1 (BOP), single–port, 32–11
appended to transmit string, 10–6
defined for BOP, 10–6, 37–11, 38–14, 43–9, 44–8, 46–7, 47–7
field on BCC Setup menu, 10–13
monitor/receive condition, Layer 1 (BOP only), 32–4
overlay, BOP only, 10–4

Absolute pathnames, files and directories, 15–6

Accumulate, layer–independent action, 31–15

Accumulate action, 21–7
may apply to current, last, minimum, or maximum values, 21–9
not found on trigger menus, 21–8
used to log one hour per day over days or weeks, 21–10

Accumulate counter, layer–independent action, 31–15

Accumulate timer, layer–independent action, 31–15

Accumulator
created by being named in Accumulate action, 31–15
may be given display line on stat screen(s), 31–15
printing line of tabular statistics for, 31–16

Actions
capture memory on/off, 7–11
Enhance, control of color display, 18–5
Layer 1
accessed via Done key, 32–8
dual–port, 32–17
dual–port, 32–17

single–port, 32–8
layer–independent, 31–8–31–13
Protocol Spreadsheet, programming block, 28–9
comments in, 28–15
record on/off, 7–11

ADDR
address, trace column
SDLC, 39–8
X.25 Layer 2, 37–7
field on SDLC Frame Level Setup screen, 39–3, 39–5, 40–3

Address
adjunct to monitor/receive–frame condition
SDLC, 39–14
X.25 Layer 2, 37–14
adjunct to send–frame action
SDLC, 39–24
X.25 Layer 2, 37–25
trace field, SNA–SDLC, 40–7

Affects, field on BCC Setup menu, 10–13

Again, editor command, 30–9

AIM Interface Setup, overview of screen, 2–12

Alarm
field on Trigger Setup menu, 26–11
layer–independent action, 31–13

Allocating disk space, 14–3

ANSI format, SS#7 layer 3, 47–4

Append, run–mode printer output to existing disk file, 16–6

ASCII, default BCC parameters, 10–9

ASCII keys, in programming menus and spreadsheet, 3–4

Async
data setup, 5–15
sample Line Setup, 5–18

ATM, 49–1
AAL protocols, 49–3
ATM Adaptation Layer (AAL), 49–2
ATM header, AAL 1, 49–7
ATM Layer, 49–2
cell relay, 49–1
CPCS header, AAL 3/4, 49–9
CPCS trailer
AAL 3/4, 49–9
AAL 5, 49–15
framing
HEC, 49–2
PLCP, 49–2
HEC framing, 49–1, 49–18
OAM cells, 49–17
overview, 49–1
packet–switching features, 49–1
Physical Layer, 49–2
PLCP framing, 49–1, 49–18
recording selections, T1/E1 data, 49–1
SAR header
AAL 1, 49–5
AAL 3/4, 49–11
SAR trailer, AAL 3/4, 49–11
transmission
data, 49–2
video, 49–2
voice, 49–2

ATM Adaptation Layer (AAL), in ATM, 49–2

ATM header
AAL 3/4, 49–12
AAL 5, 49–16
in ATM, AAL 1, 49–7

ATM Layer, in ATM, 49–2

Auto Configure, screen in automonitor mode, 6–3

Auto–indent, editor command, 30–9

Automatic OSI primitives. *See* Primitives

Automatic X.25 Layer 2, 37–36

Automonitor mode
setting up, 5–3
stage in autoconfiguration displayed in Status field, 6–3
updates Line Setup screen, 6–4
with no clock, speed defaults to 168 kbps, 6–4

Autosync, subfield on Line Setup menu, 5–8, 5–12, 5–13

AUX I/O, connector, 1–11

AUX leads, 1–11

AUX outputs, on/off, Layer 1 emulate–mode action, 32–12

Auxiliary TTL connector, 1–11

Average, column on Tabular Statistics screen, 21–6, 21–11, 31–10

B

Back panel, 1–8
fan, 1–10
fan filter, clean to prevent overheating, 1–10
frequency selection, 1–9
Input/Output connectors, 1–10
10BASE–T Ethernet, 1–11

AUXILIARY TTL, 1–11
CRT/RGB, 1–11
ISDN handset, 1–10
PRINTER, 1–10
REMOTE RS–232, 1–10
RJ–45 Ethernet, 1–11
RS–170 composite video, 1–11
on/off (power) switch, 1–10
power connector, 1–8
voltage selection, 1–8

Backslash, entry of inside prompt message, 31–12

Bad, Summary subfields on Disk Maintenance menu, 14–10

Bad BCC
adjunct to monitor–frame condition
LAPD, 43–14, 43–15
SDLC, 39–15, 39–17
X.25 Layer 2, 37–14, 37–15
adjunct to send–frame action
LAPD, 43–25
SDLC, 39–27
X.25 Layer 2, 37–28
adjunct to send–string action, Layer 1, single–port, 32–11
appended to transmit string, 10–5
as condition, 10–5
monitor/receive condition, Layer 1, 32–4
operational only when Rcv Blk Chk enabled, 32–4
overlay, 10–3

Baudot, no default BCC parameters, 10–10

BCC
See also Block checking
cross between a Layer 1 and Layer 2 function, 32–4
Layer 1 condition, 32–4
operational only when Rcv Blk Chk enabled, 32–4
subfield on Trigger Setup menu, 10–5, 26–4
trace column
LAPD, 43–9
Q.931, 44–8
SDLC, 39–11
SNA–SDLC, 40–6
SS#7 Layer 2, 46–6, 47–7
X.25 Layer 2, 37–11
X.25 Layer 3, 38–14

BCC Setup, overview of screen, 2–12

BCC Setup screen, SMDS Frame Mode, 10–4

Begin, editor command, 30–5

BERT
"force–loopback" programming example, 11–21–11–22
analyze–only mode, 11–18
automatic error injection, 11–14
enabling/disabling by softkey, 11–18
status message, 11–21
block size, 11–13
clearing counters, 11–6
clearing the results screen, 11–18
counters, 11–19
five pseudorandom patterns, 11–3
algorithms, 11–3
freeze mode, 11–17
half duplex, 11–6
"receive and analyze" versus "generate" mode, 11–6
initiating the send–receive cycle, 11–7
manual error injection, 11–18
operating mode, selected on Line Setup menu, 11–5
patterns, 11–7
reinitializing a running test, 11–18
relation of BERT Setup menu to Interface Control screen, 11–7
relation of BERT Setup menu to Line Setup menu, 11–6, 11–14
results, 11–19
statistics display, 11–19
synchronous versus asynchronous, 11–15
test length, 11–14

BERT modes, setting up, 5–4

BERT Setup, overview of screen, 2–12

Binary display, of cursor characters, 6–15
in relation to order of transmission, 6–15

Bisync
CRC mode, 10–14
advantage over selectable mode, 10–14
field on BCC Setup menu, 10–12
sample Line Setup, 5–18

Bit mask
in Protocol Spreadsheet strings, 33–3
in Receive string condition, 25–7
in Suppress field, 6–11
to detect XON and XOFF only, 6–12

Bit Mask key, 3–6

Bit order
in relation to hexadecimal notation, 6–14
in relation to pattern sync in BERT, 11–11
normal versus reverse, 5–16

Bit Order/Polarity, field on Line Setup menu, 5–16, 11–15
significance in BERT testing, 11–16

Bit–image data, 7–3
  playback, 7–4

Bits
  field on Line Setup menu, 5–7, 11–15
  in BERT testing, 11–15
  number of, in setup, 5–7

Bits In Error, BERT counter, 11–20

Blnk, subfield on Trigger Setup menu, 26–7

Block
  component of BERT test, 11–13
  editor command, 30–5

Block checking
  automatically on for BOP, 10–4
  distinction between transmitting and evaluating BCC, 10–3
  enabling BCC overlays, 5–13, 10–3
  for DDCMP
    automatically on, 41–1
    data BCC may be tested as event variable in C, 41–2
    header BCC only may be tested by trigger, 41–2
  looking under BCC overlay, 10–4
  parameters defined on BCC Setup menu, 10–6
  result used as trigger condition, 10–5

Block Size, field on BERT Setup menu, 11–13

Blocks In Error, BERT counter, 11–20

Blocks Received, BERT counter, 11–20

Blocks Sent, BERT counter, 11–20

Boot–up
  creating a user interface, HRD/usr/user_intrf, 2–4
  enable Easy View, /sys/ezview_setup, 2–3
  entry into Easy View, 4–3
  from the keyboard, 2–2, 4–3
  loading Printer Setup, /sys/print_setup, 2–3, 16–7
  running default program, /usr/default, 2–6

BOP, synchronization and BCC parameters always defined for, 5–14, 10–4, 10–8

Break key, 30–5
  in Easy View, 4–16

Buffer Control Leads, field on Front–End Buffer Setup screen, 9–5

Buffer Full
  condition, Trigger Setup menus, 25–11
  layer–independent condition, 31–5

C

C, color, field on Graphical Statistics menu, 22–7

C language, third tier in programming hierarchy, 23–4

C/R
  adjunct to monitor/receive–frame condition, LAPD, 43–13
  adjunct to send–frame action, LAPD, 43–22
  Command/Response, trace column, LAPD, 43–8

CAL–REF–VAL, call reference value, trace column, Q.931, 44–5

CALL, send action, X.25 Layer 3, 38–31

Call Confirm
  send–packet action, X.25 Layer 3, sending "short" version without addresses and facilities, 38–33
  sent down (as primitive) to Layer 2, 34–9

Call reference value
  adjunct to monitor/receive–message condition, Q.931, 44–10
  monitor/receive condition, Q.931, 44–10
  trace column, Q.931, 44–5

Call Request
  as character data, 38–9
  as entry on X.25 Packet Level Setup screen, 38–6
  as packet on trace display, 38–8
  send–packet action, X.25 Layer 3, 38–32
  sent up (as primitive) from Layer 2, 34–9

Call Request user data, may be longer than ten character spaces, 38–7

CALL_CONF, send action, X.25 Layer 3, 38–31

CALLED, field on X.25 Packet Level Setup screen, 38–6

Called address, entered in CALLED field on X.25 Packet Level Setup screen, 38–6

CALLING, field on X.25 Packet Level Setup screen, 38–6

Calling address, entered in CALLING field on X.25 Packet Level Setup screen, 38–6

Capture, field on Trigger Setup menu, 26–11

Capture data to screen (on/off), Layer 1 action, single–port, 32–16

Capture Memory, field on Record Setup menu, 7–4, 7–6, 7–10, 14–10

Capture memory
  See also Data capture; Recording data

Freeze key, 3–11

Carriage Return, produced by operation of CTRL and M keys, 33–1

Cause byte
adjunct to Restart, Reset, Clear, and Reg Confirm actions, 38–36
adjunct to Restart, Reset, Clear, and Reg Confirm conditions, 38–20
listed for Reset, Clear, and Reg Confirm packets, 38–22
listed for Restart packet, 38–21
listed for Send Clear actions, 38–37

CCITT
format, SS#7 layer 3, 47–4
Open Systems Interconnection model, 24–4–24–8
See also Layers

CCSS#7. *See* SS#7

CD
available for triggering, 32–5
field on RS–232 Interface Control menu, 11–4, 11–6

CD on/off, Layer 1 Emulate DCE action, 32–12

Cell relay, for ATM, 49–1

Chaining, of programs via Load Program action, 31–19

Change Directory, File Maintenance, menu selection, 15–20

Change idle character, Layer 1 action, single–port, 32–13

Char, subfield on Line Setup menu, 5–13

Character buffer, 7–4
capacity, 6–27
data, 7–3
playback, 7–4
recording, 7–11
storage capacity, 1–15

Character data
buffer correlation with trace data, 6–27
display of
accessed by DATA softkey, 6–7
dual line, 6–8
single line, 6–8

Character field, defined, 38–7

Circuit Identifier Code (CIC), SS#7 Layer 3, 47–6, 47–10

Clear, editor command, 30–5

Clear key, 30–4
in Easy View, 4–16
in menu fields, 3–7
in spreadsheet, 3–7

Clear path, emulate–mode action, X.25 Layer 3, 38–41

Clear statistical accumulator values, 31–15

Clear statistical counter values, layer–independent action, 31–10

Clear statistical timer values, layer–independent action, 31–11

Clock
field on Line Setup menu, 11–16
in BERT testing, 11–16
signal, 1–16
See also Speed
time–of–day, 1–16
See also Date/Time Setup

Clock Source, field on Line Setup menu, 5–15

Code
field on Line Setup menu, 5–6, 10–9, 11–14, 11–15
significance in BERT testing, 11–14
standard codes, 5–6

Color display
color graphics, 18–6
connectors for external monitors, 18–3
miscellaneous utilities, 18–3–18–6
selectable options, 18–4–18–5
background color, 18–5
blink, 18–5
character, 18–5
trigger control of, 18–5–18–6

Command
field on File Maintenance menu, 3–6
in Easy View script file, 20–25

Command addressing, adjunct to send–frame action, X.25 Layer 2, 37–24

Comment, 28–14
debugging tool, 28–15
delimiters, 28–14
length of, 28–14
location on spreadsheet, 28–14–28–15
purpose of, 28–15
valid characters, 28–14

Common Channel Signalling System #7. *See* SS#7

Compilation
automatic during LPRGRM save, 15–18
automatic during object–code save, 15–18
error diagnostics, 2–23

fields that can be modified without causing recompile, 2–22
rerun without recompiling, 2–21
seven phases, 2–20

Compile, File Maintenance
compiles contents of file or spreadsheet, 15–25
compiling spreadsheet generates #pragma hooks, 15–26
menu selection, 15–25

Conditions
EIA, fails to come true, 2–26
Layer 1
dual–port, 32–6
single–port, 32–1
layer–independent, 31–3–31–7
counters in linkable–object files, 28–12
Protocol Spreadsheet
naming requirements, 31–1
programming block, 28–8
comments in, 28–15
rules for combining conditions, 31–2
transitional vs. status, 31–2, 31–6

Confirm primitives, 34–5

CONNECT IND primitive, example on spreadsheet, 34–7

CONNECT REQ primitive, example on spreadsheet, 34–8

Connection Security Timeout, Remote Control Setup, 2–17

Connection Type, Remote Control Setup, 2–17

Connectors
back panel, 1–8
power, back panel, 1–8
RGB, 18–3
RS–170 video, 18–3
RS–232 printer connector, 16–3
Test Interface Module, 1–12–1–29

Constants, 29–3–29–7
expansion of, 29–7
fox message, 33–4
in Receive string condition, 25–8
in spreadsheet string, 33–3
legal names of, 29–4
nesting of, 29–6
Protocol Spreadsheet, 24–8
programming block, 28–8
comments in, 28–14
referencing, 29–5
scope of, 29–4
transmitting, 33–4

Control characters
data–entry of, 33–1
enhancement of via bit mask, 6–13

Control leads
See also EIA leads
playback, 2–20
of bit–image data, 2–24, 9–5
of character data, 2–24, 9–5

Copy
editor command, 30–6
File Maintenance, menu selections, 15–20

Counter
accumulated, 31–15
condition
Protocol Spreadsheet, 21–4, 31–5
when used in linkable–object files, 28–12
Trigger Setup menus, 21–4, 25–11
identified by name on statistics screen, 21–4
layer–independent action, 31–8
maximum value vs. maximum stat display, 31–6
may be identified on statistics screen following run, 21–6
printing line of tabular statistics for, 31–16
relational operators, 31–5
shared between spreadsheet and Trigger Setup menus, 31–5, 31–9
transmitted, 33–4

CPCS header, in ATM, AAL 3/4, 49–9

CPCS trailer, in ATM
AAL 3/4, 49–9
AAL 5, 49–15

CR control character, 3–6

CRC Mode, field on BCC Setup menu, 10–9, 10–12, 10–14, 10–16

CRT/RGB connector, 1–11

CTS
available for triggering, 32–5
field on RS–232 Interface Control menu, 11–4, 11–6

CTS on/off, Layer 1 Emulate DCE action, 32–12

Current, column on Tabular Statistics screen, 31–10, 31–11

Current Date, field on Date/Time Setup menu, 17–3

Current directory
File Maintenance screen, 15–13
filing system, 15–4

Current Time, field on Date/Time Setup menu, 17–3

Cursor keys
in Easy View, 4–7, 4–16

in spreadsheet, 3–9
may be programmed in the Display Window, 3–12, 6–23
on menu screens, 3–8
used to control playback speed, 3–12

Cursor timing
in data–plus–leads displays, 6–16
Mark key, 6–16

D

D, trace column, X.25 Layer 3, 38–13

D bit
adjunct to monitor/receive–packet condition, 38–18
adjunct to send–packet action, 38–36
position diagrammed, 38–13
value selectable for Call and Call Confirm packets as well as Data, 38–19

DAT, source of playback data from disk, 5–5

DAT Record Size, field on Record Setup menu, 7–7

DATA, field on X.25 Packet Level Setup screen, 38–7

Data
See also Character data
ATM transmission, 49–2
bit–image data, 7–3
buffered automatically in FEB, 9–3
character–oriented, 7–3

Data acquisition tracks, 14–3

Data capture, 2–24
See also Playback; Recording data
manual control of, 7–12
RAM, data storage, 1–15
trigger control of, 7–11

Data compression, SS#7, Layer 1, 45–4

Data display, black and white enhancements, 18–6

DATA IND primitive, example on spreadsheet, 34–9

Data packet
monitor/receive condition, X.25 Layer 3, 38–15
send action, X.25 Layer 3, 38–31

Data playback, control of, 7–10

Data plus leads
cursor timing, 6–16
display available during playback, 9–5
display enabled/disabled by FEB setup, 9–5
display of, 6–9
control leads selected for, 6–9
softkey access, 6–9
failure of leads to transition, 2–27

DATA REQ primitive, example on spreadsheet, 34–9

Data source, connection to, 1–18

Data suppression, SS#7, Layer 1, 45–5

Data to Record, field on Record Setup menu, 7–7

Data Transfer, Disk Maintenance, menu selection, 14–10

Data transfer
INTERVIEW 5, 10, 15 PLUS data, 14–13–14–19
prior to playback, 7–4

Date/Time Setup, 17–3
menu selections, 17–2
set date, 17–3
set time, 17–3

Day of month, as trigger condition, 31–6

DCE, monitor condition
LAPD, 43–9
Layer 1, 32–3
Q.931, 44–9
SDLC, 39–12
SS#7 Layer 2, 46–7, 47–7
X.25 Layer 2, 37–11
X.25 Layer 3, 38–15

DCE2, monitor condition, Layer 1, 32–8

DDCMP, Layer 1 package, 41–1

Decimal field, defined, 38–6

Decrement counter, layer–independent action, 31–10

Decrement flag byte, as 16–bit binary counter, layer–independent action, 31–14

Default menus, how to change, 2–6

Delete
editor command, 30–5
File Maintenance, menu selection, 15–25

Delete Char key, 3–7, 30–4
in Easy View, 4–16

Delete Line key, 3–7, 30–4

Descript, Summary subfields on Disk Maintenance menu, 14–9

Destination Point Code (DPC), SS#7 Layer 3, 47–10

Diagnostic byte
adjunct to Restart, Reset, Clear, Diag, and Reg Confirm conditions, 38–23

adjunct to Restart, Reset, Clear, Diag, and Reg Confirm send actions, 38–38
entered as two hex digits, 38–23

Directories
/sys, 15–6
/usr, 2–4, 15–6
/usr/default, 2–6
absolute pathnames, 15–6
directory listings, 15–13
filing system, how to create, 15–5
naming conventions, 15–7
relative pathnames, 15–6
root (/) directory, 15–4
write–protected, 15–13

DISABLE, mode on Line Setup screen, 5–5

Disk drives, 1–5
current disk, filing system, 15–12
drive references and priority, 14–3
filing systems, moving from disk to disk, 15–5
microfloppies
compatibility, 1–6
write protection, 1–6
microfloppy disks, storage capacity, 1–5
Winchester hard disk, 1–7

Disk Maintenance, 14–3–14–19
allocating disk space, 14–3
command
Data Transfer, 14–10
Disk Summary, 14–9
Duplicate Disk, 14–12
Format Disk, 14–5
INT 10, 14–13–14–19
data acquisition tracks, 14–3
data transfer, 15–7
description of disks, 14–3
initializing system, INTERVIEW 7000, 2–3
installing new system software, 2–7
menu selections, 14–2, 14–4–14–20
overview, 2–15

Disk Name, subfield on Disk Maintenance menu, 14–6

Disk No
field on Line Setup menu, 7–4
field on Record Setup menu, 7–6, 7–10

Disk Number, subfield on Disk Maintenance menu, 14–5

Disk Summary, Disk Maintenance, menu selection, 14–9

Display, electroluminescent (EL), 1–4

Display Abort
field on Line Setup menu, 25–6
subfield of BOP Format, 5–14
subfield on Line Setup menu, 5–14

Display Idle
field on Line Setup menu, 32–13
subfield on Line Setup menu, 5–13, 6–10, 9–3, 9–4
cannot display idle if suppressed in FEB, 9–4

Display Mode, field on Display Setup menu, 6–7, 6–18

Display Program Warning Messages, Easy View Setup menu field, 19–4

Display Setup
menu selections, 6–2
traces, 2–27

Display Setup screen, overview, 2–11

Display States, field on Display Setup menu, 6–20, 31–19

Display Window
cursor keys under programmer's control, 6–23
display mode, 6–23
softkey labels under programmer's control, 6–24

DL data, 34–10

DL_CONNECT CONF
entered manually at Layer 2 to "fool" Layer 3 into thinking there is a link, 38–46
primitive forced up by user program at Layer 2, 34–6
primitive sent upward by Layer 2 to confirm the link, 34–9

DL_CONNECT IND
action primitive at Layer 2, 34–3
condition primitive at Layer 3, 34–3

DL_CONNECT REQ
automatic when data primitives are passed down by Layer 3, 35–1
primitive passed down from Layer 3, 34–8
primitive triggered automatically by Layer 3 Send action, 34–6
sent down automatically at Layer 3 if Layer 2 inactive, 38–44, 38–46

DL_DATA
macro, 33–5
primitives between Layers 2 and 3, 34–9

DL_DATA IND
condition at Layer 3, 43–31
sent up automatically by Give Data action at Layer 2, 37–28, 37–34, 39–28, 39–34, 43–25, 43–31

DL_DATA REQ, sent down automatically by Send or Resend action at Layer 3, 38–44

Don't Care key, 3–6
in Receive string condition, 25–7

Done key
on menu screens, 3–10
used to change real–time display softkeys, 6–6
used to exit softkey rack in spreadsheet, 3–10
used to move from Conditions to Actions, 37–21, 38–29, 39–22, 43–19

Double parens, 3–5
in Protocol Spreadsheet string, 33–3
in Receive string condition, 25–8

Down Arrow key, 30–4

Drive, field on Layer Setup screen, 8–4

Drive Type, subfield on Disk Maintenance menu, 14–6

DSR, available for triggering, 32–5

DSR on/off, Layer 1 Emulate DCE action, 32–12

DTE, monitor condition
LAPD, 43–9
Layer 1, 32–3
Q.931, 44–9
SDLC, 39–12
SS#7 Layer 2, 46–7, 47–7
X.25 Layer 2, 37–11
X.25 Layer 3, 38–15

DTE2, monitor condition, Layer 1, 32–8

DTR, available for triggering, 32–5

DTR on/off, Layer 1 Emulate DTE action, 32–12

Dual–Port directive, #pragma quad_display, creates QUAD softkey, 6–6

Dual–Port DS–0A, optional TIM, 13–1

Dual–Port G.703, optional TIM, 13–1

Dual–Port G.703 Co–Directional, optional TIM, 13–1

Dual–Port T1, optional TIM, 13–1

Dual–Port V.35/EIA–530(449)/EIA–232, optional TIM, 13–1

Duplicate Disk
Disk Maintenance, menu selection, 14–12
installing new system software via the DUPDISK command, 2–7

DXI, SMDS interface, 48–1

E

Easy View
default in TURBO units after power–up, 4–3
display warning messages, 19–4
enable/disable, 19–3
enter after power–up, 19–4
EZ VU key, 4–4
installing updates, 4–19
keep menu info in memory, 19–4
keys used in, 4–7, 4–14
load program edit timeout, 19–5
menus
format, 4–5
help information about selections, 4–7
item description, 4–6
keys, 4–6, 4–7
levels, 4–4
master title, 4–5
changing the title, 20–22
menu level, 4–5
menu title, 4–5
changing the title, 20–23
selections, 4–6
tree–structed hierarchy, 4–4
mstrmenu.txt, format, 20–20
overview, 4–3
printing text files, 4–15
screen saver timeout, 19–5
script file
commands, 20–25
format, 20–20
help–file pathname, 20–25
item date, 20–24
item description, 20–25
item name, 20–24
labels, 20–23, 20–25
master information, 20–21
master title, 20–22
master title date, 20–22
menu date, 20–24
menu information, 20–23
menu title, 20–23
menu–item information, 20–24
sample, 20–27
setup menu, 4–3
uses
access help or tutorial information, 4–3
quickly load and run programs, 4–3

Easy View Setup, 4–3
overview, 2–15

Easy View Setup screen, menu selections, 19–2, 19–3

Easy View system. *See* Easy View

EBCD
  default BCC parameters, 10–10
  reverse bit order appropriate for, 5–16

EBCDIC, default BCC parameters, 10–9

Edit key, 3–7

Editing keypad, 30–3

Editor, Protocol Spreadsheet
  See also Protocol Spreadsheet editor
  function keys, 30–5–30–10

EIA
  condition, Trigger Setup menus, 25–9
  Layer 1 conditions, 32–5
  Layer 1 emulate–mode actions, single–port, 32–12
  trigger conditions, fails to come true, 2–26

EIA leads
  buffered or discarded in FEB, 9–3
    effect on character–buffer capacity, 6–27
    effect on data–plus–leads display, 9–5
    effect on EIA trigger conditions, 9–3
  maintaining lead statuses during program chaining, 31–19
  storage of, 1–15

Emulate
  field on LAPD Frame Level Setup screen, 43–3, 43–4
  field on SDLC Frame Level Setup screen, 39–3, 39–4
  field on SNA/SDLC Frame Level Setup screen, 40–3
  field on X.25 Frame Level Setup screen, 37–3, 37–4, 37–24
  field on X.25 Packet Level Setup screen, 38–3, 38–4

Emulate modes, setting up, 5–4

Emulation, connectors used, 1–12

Emulation Addressing, field on SDLC Frame Level Setup screen, 39–3, 39–5, 40–3

Enable Easy View, Easy View Setup menu field, 19–3

Enable Watchdog, Remote Control Setup, 2–16

End, editor command, 30–5

End/Incl, field on BCC Setup menu, 10–13

End/N/Incl, field on BCC Setup menu, 10–13

End/Staystarted/Incl, field on BCC Setup menu, 10–13

End/Staystarted/N/Incl, field on BCC Setup menu, 10–13

Enhance
  field on Display Setup menu, 6–12, 7–5
  field on Miscellaneous Utilities menu, 18–4, 18–5
  field on Trigger Setup menu, 26–7

Enhance character data
  as Layer 1 action, single–port, 32–14
  on Display Setup, 6–12

Enhance selected trace rows
  LAPD action, 43–29
  map to color display, 37–33, 38–42, 39–32, 43–30, 44–12, 46–10, 47–11
  Q.931 action, 44–12
  SDLC action, 39–32
  SNA action, 40–3
  SS#7 Layer 2 action, 46–10
  SS#7 Layer 3 action, 47–11
  X.25 Layer 2 action, 37–32
  X.25 Layer 3 action, 38–42

Enhancements
  black and white, 18–6
  color, 18–3–18–6
  low intensity, 32–15
  must be turned off as well as on at Layer 1, 32–14
  used to highlight Bisync addresses, 32–14

Enter Easy View After Power–Up, Easy View Setup menu field, 19–4

Enter State, layer–independent condition, 31–3

Error Injection Rate, field on BERT Setup menu, 11–14, 11–22

Error–free seconds, BERT counter, 11–21

Errors
  in BERT
    automatic injection, 11–14
    manual injection, 11–18
  recoverable, 2–23

Ethernet network
  Gateway Address, Remote Control Setup, 2–19
  Host Address, Remote Control Setup, 2–19
  Subnet Mask, Remote Control Setup, 2–19

Execute key, 3–7

External monitors
  control of enhancements
    black and white, 18–6
    color, 18–3–18–6
  RGB color video connector, 1–11, 18–3
  RS–170 video connector, 1–11, 18–3

EZ VU key, 3–4, 3–6, 4–4, 4–8

F

FACILITIES, field on X.25 Packet Level Setup screen, 38–7

Facilities
adjunct to send–call action on Protocol Spreadsheet, X.25 Layer 3, 38–32
relation to FACILITIES field on X.25 Packet Layer Setup screen, 38–32
length byte handled automatically, 38–7

Fan
back panel, 1–10
clean filter to prevent overheating, 1–10

Fault, in half–duplex BERT, 11–9
on noisy circuit, 11–9

FEB. *See* Front end buffer

FEB Setup, overview of screen, 2–12

File, source of playback data from disk, 5–5

File Maintenance, 15–3–15–20
absolute pathnames, 15–6
creating new directories, 15–5
current directory, 15–13
default directory, 15–4
directories, 15–4
File Maintenance screen, 15–11
current disk, 15–12
directory listings, 15–13
overview, 2–14
filenames, using wildcards, 15–8
files
data files, 15–7
description of, 15–6
linkable program files, 15–7, 15–16
linkable object files, 15–7, 28–10
compiled contents of spreadsheet, 28–10
loading and saving, 15–3
marking files, 15–15
moving from file to file, 15–4
object files, 15–7, 15–16
program files, 15–6, 15–16
Protocol Spreadsheet, 15–7
selecting files for command execution, 15–14
setup files, 15–6, 15–16
types, 15–14
unmarking files, 15–15
menu selections, 15–2
Change Directory, 15–20
Compile, 15–25
compiles contents of file or spreadsheet, 15–25
compiling spreadsheet generates #pragma hooks, 15–26
Copy, 15–20
Delete, 15–25
how to execute, 15–16
Load, 15–16
Make Directory, 15–20
Print, 15–23
Rename, 15–24
Save, 15–17
View, 15–23
Write Enable, 15–24
Write Protect, 15–24
moving from disk to disk, 15–5
naming conventions, files and directories, 15–7
pathnames, the use of periods, 15–10
relative pathnames, 15–6
root (/) directory, 15–4
the /sys directory, 15–6
the /usr directory, 15–6
write–protected files, 15–13

Find, editor command, 30–8

Flag key, 3–6
in Receive string condition, 25–7
valid in Suppress field, 6–11

Flags
common to all tests and layers, 31–13
condition, Trigger Setup menus, 25–10
layer–independent action, 31–13
as toggling mechanism, 31–14
layer–independent condition, 31–7
transmitted, 33–4

FLG, flag, trace column, Q.931, 44–5

Force data–packet transmit, 38–45

Force receivers out of sync, Layer 1 action, single–port, 32–13

Format, field on Line Setup menu, 5–7, 5–13, 11–15, 25–6, 32–4, 32–14
significance in BERT testing, 11–15

Format Disk, Disk Maintenance, menu selection, 14–5

Fox message, 32–10, 32–18, 37–27, 38–38, 39–27, 43–25
"forced down" from Layer 3 to the Layer 1 interface, 34–6
in BERT, 11–7
really a built–in constant, 33–4

Frame fields
diagrammed for LAPD, 43–7

diagrammed for SDLC, 39–9
diagrammed for X.25, 37–8

Frame Mode, 5–14

Frame mode, optimizing high–speed performance, 2–34

Frame sent, emulate–mode condition
LAPD, 43–17
SDLC, 39–19
should be used along with More/No More to Resend, 37–20, 39–21, 43–18
X.25 Layer 2, 37–18

Frame Timestamps, field on Front–End Buffer Setup screen, 9–8, 37–10, 38–14, 39–11, 40–6, 43–9, 44–8, 46–6, 47–7

Framing
HEC, in ATM, 49–2
PLCP, in ATM, 49–2

Framing error, monitor/receive condition, Layer 1, 32–4

Free, Summary subfields on Disk Maintenance menu, 14–10

Freeze key, 3–11
contrasted with Capture On/Off trigger action, 3–11

Freeze LED, front panel, 1–5

Freeze mode
in BERT, 11–17
initiated by trigger. *See* Capture data to screen (on/off)
parallel cursor movement during, 6–27, 37–6, 38–8, 39–7, 40–4, 44–4, 46–4, 47–4

Frequency selection, back panel, 1–9

From
field on Disk Maintenance menu, 7–4
subfield on Disk Maintenance menu, 14–10

From Disk Number, subfield on Disk Maintenance menu, 14–12

Front end buffer
applies to playback of bit–image data, 2–24
Idle Suppress field, does not apply to playback of bit–image data, 2–24
on playback path of bit–image data, 2–24, 9–5, 9–6
setup
effect on character–buffer capacity, 6–27
effect on data–plus–leads display, 6–9
time ticks and EIA leads, storage of, 1–15, 2–24

Front End Buffer Setup, menu selections, 9–2

Front panel, 1–3
electroluminescent (EL) display, 1–4
function keys, 1–4
LED overlay, 1–5
LED's, 1–5, 2–24
U/A, 1–13

Function keys, 1–4

G

G.703
optional TIM, 13–1
record setup, 7–8

Gateway Address, Ethernet network, Remote Control Setup, 2–19

General operation, 2–1–2–34
boot–up program, creating a user interface, 2–4
changing default menus, 2–6
common problems, 2–26–2–30
data capture, 2–24
front end buffer, 2–24
front end buffer, on playback path of bit–image data, 2–24
initializing system, INTERVIEW 7000, 2–3
installing new system software, 2–7
overview of menus, 2–9–2–19
power up, 1–19, 2–1
rerunning a test program, 2–21
running a test program, 2–20
running default program, 2–6

Give data, 33–5
LAPD, 43–25
SDLC action, 39–28
X.25 Layer 2 action, 34–10, 37–28
X.25 Layer 3 action, 38–39

Go–error, editor command, 30–10

Go–line, editor command, 30–9

Good BCC
adjunct to monitor–frame condition
LAPD, 43–14, 43–15
SDLC, 39–15, 39–17
X.25 Layer 2, 37–14, 37–15
adjunct to send–frame action
LAPD, 43–25
SDLC, 39–27
X.25 Layer 2, 37–28
adjunct to send–string action, Layer 1, single–port, 32–10
appended to transmit string, 10–5
as condition, 10–5
default BCC for frames, 37–28, 39–27, 43–25

monitor/receive condition, Layer 1, 32–4
operational only when Rcv Blk Chk enabled, 32–4
overlay, 10–3
parameters defined on BCC Setup menu, 10–6

Graphical Statistics menu, color graphics, 18–6

H

Half–duplex BERT, 11–6

Handset connector, 1–10

Handshake, field on BERT Setup menu, 11–5, 11–6, 11–7, 11–12
relation to Interface Control screen, 11–7

Hardware, 1–3–1–19
back panel, 1–8
clock, 1–16
disk drives, 1–5
front panel, 1–3
keyboard, 1–3
operating environment, 1–16
operating positions, 1–17
physical dimensions, 1–3
power up, 1–18
storage capacity, 1–15

Header Error Control. *See* HEC framing

HEC framing
for ATM, 49–1, 49–18
in ATM, 49–2

Help file, pathname, in Easy View script file, 20–25

Hex, subfield on Trigger Setup menu, 26–7

Hex key
for hexadecimal data entry, 3–6
for hexadecimal translation of line data, 6–14
LED display on keycap, 6–14

Hexadecimal display
in relation to order of transmission, 6–14
turned on/off by trigger action, 6–15

Hexadecimal field, defined, 38–6

High Outgoing Channel #, field on X.25 Packet Level Setup screen, 38–3

High speed
data recording, 7–8
optimizing performance, 2–29, 2–30, 2–31
frame mode, 2–34

High Speed Frame Mode, subfield of BOP Format, 5–14

Home key, 3–9, 30–4
in Easy View, 4–8, 4–14

Host Address, Ethernet network, Remote Control Setup, 2–19

Host Port, SNA frame setup selection, 40–3

HRD/usr/user_intrf
affect on Start Up screen, 2–3
boot–up program, creating a user interface, 2–4

HS Frame Mode, subfield on Line Setup menu, 5–9

I

I, intensity, field on Graphical Statistics menu, 22–7

Idle
change idle–line character, 32–13
display in relation to Outsync action, 32–13
display used for visual record of time intervals, 9–4
displaying for visual record of lead timings, 6–10
displaying synchronous idle, 5–13
retained/discarded on FEB Setup menu, 9–3, 9–4
selecting transmit idle, 5–14
voltage not affected by inverted polarity, 5–17

Idle line action, used in X.21 bis, 36–4

Idle Suppress, field on Front–End Buffer Setup screen, 9–4
does not apply during playback, 2–24

Idle Timeout
conditions under which timer expires, 39–4
expired, emulate–mode condition, SDLC, 39–19
field on SDLC Frame Level Setup screen, 39–3
field on SNA/SDLC Frame Level Setup screen, 40–3
maximum and minimum values, 39–4

IL buffer
See also IL BUFS
and primitives, 34–3
configuring number/size of, via IL BUFS identifier, 28–13
created by DDCMP package, 41–2
default number/size of, 28–13
string to be referenced in, 34–6

IL BUFS
See also IL buffer

Protocol Spreadsheet, programming block, 28–8, 28–13
comments in, 28–14

In/out, editor command, 30–6

Increment counter, layer–independent action, 31–9

Increment flag byte, as 16–bit binary counter, layer–independent action, 31–14

Indication primitives, 34–5
versus "Requests", 34–10

Info element, Q.931, 44–5

INFO frame
monitor/receive condition
LAPD, 43–10
SDLC, 39–13
X.25 Layer 2, 37–12
send action
LAPD, 43–20
SDLC, 39–23
used to convey DL data sent down from Layer 3, 37–23, 43–20
X.25 Layer 2, 37–22

INFO–ELEMENT, trace column, Q.931, 44–5

Initial Condition, field on Record Setup menu, 7–10

Initial State, field on BCC Setup menu, 10–12

Initializing system software, INTERVIEW 7000, 2–3

Injection Rate
field on BERT Setup menu, 11–19
status field on BERT results screen, 11–21

Input/output connectors, back panel, 1–10
AUXILIARY TTL, 1–11
CRT/RGB, 1–11
ISDN handset, 1–10
PRINTER, 1–10
REMOTE RS–232, 1–10
RJ–45 Ethernet, 1–11
RS–170 composite video, 1–11

Insert Char key, 30–4
in menu fields, 3–7
used to exit insert mode, 3–8

Insert Line key, 30–4
in spreadsheet, 3–7
on statistics screens, 3–7

Insert mode, 3–8, 30–4

Installing new system software, on hard disk, 2–7

Instrument, menu field on Disk Maintenance screen, for INTERVIEW 10 transfer, 14–17

INT 10, Disk Maintenance selection, 14–17–14–19

Integrated Services Digital Network (ISDN), SS#7 Layer 3, 47–22

Interface Control Setup, overview of screen, 2–12

Interlayer buffer. *See* IL buffer

Interlayer message buffer. *See* IL buffer

Interrupt packet, sample program to enhance all, 38–43

INTERVIEW transfer, INTERVIEW 5, 10, 15 PLUS data, transfer from Disk Maintenance screen, 14–13–14–19

Invalid frame
defined, 37–7, 39–8, 43–8
receive condition
LAPD, 43–15
SDLC, 39–17
X.25 Layer 2, 37–16

Invalid packet
defined, 38–12
receive condition, X.25 Layer 3, 38–26

Invert BCC, field on BCC Setup menu, 10–12

IPARS
default BCC parameters, 10–9
default sync pattern, 5–8
reverse bit order appropriate for, 5–16
SY characters inappropriate for, 5–8

ISDN
See also Q.931
handset connector, 1–10
ISDN_D, 42–3
sample Line Setup, 5–18

ISDN S/T/U, optional TIM, 13–1

Isoc
data setup, 5–15
format in BERT, 11–15

Item date, in Easy View script file, 20–24

Item description, in Easy View script file, 20–25

Item name, in Easy View script file, 20–24

K

Keep Easy View Menu Information in Memory, Easy View Setup menu field, 19–4

Keyboard, 1–3
condition, Trigger Setup menus, 25–12

editing keypad, 30–3
function keys, 1–4
layer–independent condition, 31–4
programming keys, 3–3–3–5
real–time keys, 3–10–3–12
soft (function) keys, 3–3
for editing, 30–5–30–10

Keyword, 33–3

L

L, label, field on Graphical Statistics menu, 22–4

Labels, in Easy View script file, 20–23, 20–25

LAPD, 43–3
diagram of frame fields, 43–7
send actions, 43–19
used with ISDN D channel, 42–3

LAPD Frame Level Setup screen, 43–2

Last, column on Tabular Statistics screen, 21–6, 31–10, 31–11

Layer, field on Display Setup menu, 6–18, 6–20, 31–19

Layer 2, user program to force packets up to Layer 3 and down to Layer 1, 38–24

Layer Setup, 8–3–8–7
how to save, 8–5
Personality packages, 8–3
reside on user and hard disks, 8–3
Protocol Configuration screen, 8–4
protocols, select and load, 8–3

Layer Setup screen, overview, 2–14

Layers
identified on Program Trace, 6–20
passing data between, 2–27
program model, 24–3–24–8
Protocol Spreadsheet, programming block, 28–8
comments in, 28–14

LCN
adjunct to monitor/receive–packet condition, X.25 Layer 3, 38–18
allocation sequence, 38–5, 38–31
assigned dynamically on per–call basis, 38–4
column on X.25 Packet Level Setup screen, 38–4, 38–5
predefined for a particular call address ("path"), 38–4, 38–6
trace column, X.25 Layer 3, 38–12

LED's
front panel, 1–5, 2–24
interface status, 1–5
INTERVIEW status, 1–5
U/A, 1–5, 1–13
green–red characteristics not affected by logic (polarity), 5–17
Test Interface Module, 1–15

Left Arrow key, 30–4

Level 1 Intercha Timeout, Remote Control Setup, 2–18

Level 2 Frame Timeout, Remote Control Setup, 2–18

/lib directory, filing system, 15–27, 28–11

Line data, data capture, 2–24

Line number, of cursor position in frozen Program Trace, 6–21

Line Setup, 5–3–5–17
significance in BERT testing, 11–14

Line Setup screen
disabled, 5–5
menu selections, 5–2
Async, 5–18
Bisync, 5–18
ISDN, 5–18
SNA, 5–18
SS#7, 5–18
X.25, 5–18
overview, 2–11

Line utilization, programming example, 21–11

Linkable object files, 15–7
accessed via Object block–identifier, 28–10
advantage over object files, 15–25
C code
contents of LOBJ files, 15–7
must be compatible with menu selections, 15–26
compiled spreadsheet
accessed via OBJECT block–identifier only, 15–26
contents of LOBJ files, 15–7, 15–26
must be a valid program, 15–26
contents of, 15–25, 28–10
counters or flags in, 28–12
efficiently use memory and spreadsheet, 28–12
in /lib directory, 15–27
search rules for, 28–11
transparent to unit configuration, 15–26

Linkable–program files, 15–7
C code, in LPGM files, 15–7

changing setups, 4–17

Lists, on Protocol Spreadsheet, 33–1

Load, File Maintenance, menu selections, 15–16

Load key, 3–6, 15–16

Load program, layer–independent action, 31–19

Load Program Edit Timeout, Easy View Setup menu field, 19–5

Local Site Name, Remote Control Setup, 2–16

Logical DTE/DCE
contrasted to physical DTE/DCE, 37–4
determines command and response addresses, 37–4
determines order (ascending/descending) of LCN selection, 38–4

Loop–back C/R bit, adjunct to send–frame action, LAPD, 43–22

Loop–back P/F bit
adjunct to resend–frame action
LAPD, 43–27
X.25 Layer 2, 37–30
adjunct to send–frame action
LAPD, 43–22
SDLC, 39–25
X.25 Layer 2, 37–25

Low, subfield on Trigger Setup menu, 26–7

Low Outgoing Channel #, field on X.25 Packet Level Setup screen, 38–3

LRC Parity, field on BCC Setup menu, 10–12

LU 6.2, SNA selection, 40–3

M

M, trace column, X.25 Layer 3, 38–14

M bit
adjunct to monitor/receive–packet condition, 38–18
adjunct to send–data–packet action, 38–36
position diagrammed, 38–14

m_lo_ph_prmtv, signalled by DDCMP package, 41–2

Make Directory, File Maintenance, menu selection, 15–20

Mark key, 30–5
cursor timing, 6–16
in Easy View, 4–15, 4–17
in Freeze mode displays, 3–14
on File Maintenance screen, 3–7, 15–15
used as program tab in spreadsheet, 3–9

Marking text, in Easy View text files, 4–15

Master information, in Easy View script file, 20–21

Master title
Easy View, 4–5
changing the title, 20–22
in Easy View script file, 20–22

Master title date, in Easy View script file, 20–22

Max Level 2 Xmit Attempts, Remote Control Set-up, 2–18

Maximum, column on Tabular Statistics screen, 21–6, 31–10

Maximum data rates, data recording, 1–15

Maximum recording times, SCSI drives, 7–5

Memory, capacity, 1–15

Menu date, in Easy View script file, 20–24

Menu information, in Easy View script file, 20–23

Menu level, Easy View, 4–5

Menu selections, Easy View, 4–6

Menu title
Easy View, 4–5
changing the title, 20–23
in Easy View script file, 20–23

Menu–item information, in Easy View script file, 20–24

Menus
overview, 2–9–2–19
See also Separate listing, each menu name
configuring menus, 2–10
Program Menu, 2–9
Printer setup, 16–3–16–5
Record Setup, 7–5–7–12

Message Buffer, field on BERT Setup menu, 11–10

Message fields, diagrammed for Q.931, 44–6

Message Signal Units (MSU's), Layer 3, 47–6

Message type, Q.931, 44–5

Microfloppy disks
compatibility, 1–6
storage capacity, 1–15
write protection, 1–6

MIL, field on Line Setup menu, 5–17

MIL–188, 5–17

Minimum, column on Tabular Statistics screen, 21–6, 31–10

MISC, trace column, X.25 Layer 3, 38–14

Miscellaneous Utilities
  overview, 2–15
  with color mapping options, 18–3–18–6
    See also Graphical statistics menu, color graphics

Miscellaneous Utilities screen
  black and white enhancements, 18–6
  color display, selectable options, 18–4–18–5
    background color, 18–5
    blink, 18–5
    character, 18–5
  controlling color displays from, 18–3–18–6
  menu selections, 18–2

MOD 128, 37–4, 38–4, 39–4, 43–4

MOD 8, 37–4, 38–4, 39–4, 43–4

Mode
  field on Line Setup menu, 5–3, 7–4, 11–5, 11–6, 11–22, 32–12, 38–24, 43–14
  test mode field on Line Setup menu, 1–13

Mode of Operation
  field on Frame Level Setup screen, 38–16, 43–10
  field on LAPD Frame Level Setup screen, 43–3, 43–4
  field on SDLC Frame Level Setup screen, 39–3, 39–4
  field on SNA/SDLC Frame Level Setup screen, 40–3
  field on X.25 Frame Level Setup screen, 37–3, 37–4
  field on X.25 Packet Level Setup screen, 38–3, 38–4

Modem connector, external, 1–10

Modem Script, Remote Control Setup, 2–19

Modem script, customizing, 2–17

Monitor mode, setting up, 5–3

More to resend, emulate–mode condition
  LAPD, 43–18
  SDLC, 39–20
  X.25 Layer 2, 37–19
  X.25 Layer 3, 38–28

Move, editor command, 30–6

MPM errors, 2–23

MSG–TYPE, trace column, Q.931, 44–5

mstrmenu.txt, format, 20–20

Multi–drop
  SDLC
    enabled on Frame Level Setup screen, 39–3, 39–5
    resend frame, 39–30
    reset Nr, 39–31
    reset Ns, 39–31
  SNA, enabled on Frame Level Setup screen, 40–3

N

N, name, field on Graphical Statistics menu, 22–6

N_DATA, macro, 33–5

N_DATA IND, sent up automatically by Give Data action at Layer 3, 38–39, 38–44

National Format, field on SS#7 Packet Level Setup screen, 47–3

Negative exponent, meaning in BERT formulas, 11–14

Network Indicator, SS#7 Layer 3, 47–9

Network Management (NETM) Headers, SS#7 Layer 3, 47–18

Next Page key, 30–4
  in Easy View, 4–8, 4–14
  in spreadsheet, 3–9
  on statistics screens, 3–9
  on trace display, 37–6, 38–8, 39–6, 40–4, 44–4, 46–4, 47–4

Next state, Protocol Spreadsheet, programming block, 28–9
  comments in, 28–15

No BCC, appended to transmit string, 10–6
  interpreted as bad BCC, 10–6

No display, display mode, 6–28

No more to resend, emulate–mode condition
  LAPD, 43–18
  SDLC, 39–20
  X.25 Layer 2, 37–19
  X.25 Layer 3, 38–28

Not Equal key, 3–6
  in Receive string condition, 25–7
  used in Suppress field to indicate "display only", 6–11

Nr
  acknowledging last Ns, adjunct to send–frame action
    LAPD, 43–23
    SDLC, 39–25

X.25 Layer 2, 37–26
calculated automatically, adjunct to send–frame action
LAPD, 43–23
SDLC, 39–25
X.25 Layer 2, 37–26
repeating last Nr, adjunct to send–frame action
LAPD, 43–23
SDLC, 39–25
X.25 Layer 2, 37–26
reset, emulate–mode action
LAPD, 43–28
SDLC, for a specific controller address, 39–31
X.25 Layer 2, 37–31
trace column
LAPD, 43–8
SDLC, 39–10
staggered to indicate two separate numbering sequences, 37–9, 39–10
X.25 Layer 2, 37–9
tracked for specified addresses in SDLC multi–drop, 39–5
value, adjunct to send–frame action
LAPD, 43–23
SDLC, 39–25
X.25 Layer 2, 37–26

Nr error, emulate–mode condition
LAPD, 43–16
SDLC, 39–19
X.25 Layer 2, 37–18

NRZI, field on Line Setup menu, 5–17

Ns
calculated automatically, adjunct to send–I–frame action
LAPD, 43–24
SDLC, 39–27
X.25 Layer 2, 37–27
reset, emulate–mode action
LAPD, 43–28
SDLC, for a specific controller address, 39–31
X.25 Layer 2, 37–31
same as last–received Nr, adjunct to send–I–frame action
LAPD, 43–24
SDLC, 39–26
X.25 Layer 2, 37–27
skip to correct Ns plus one, adjunct to send–I–frame action
LAPD, 43–24
SDLC, 39–27
X.25 Layer 2, 37–27
trace column
LAPD, 43–8
SDLC, 39–10
staggered to indicate two separate numbering sequences, 37–9, 39–10
X.25 Layer 2, 37–9
tracked for specified addresses in SDLC multi–drop, 39–5
value, adjunct to send–I–frame action
LAPD, 43–23
SDLC, 39–26
X.25 Layer 2, 37–26

Ns error, emulate–mode condition
LAPD, 43–16
SDLC, 39–18
X.25 Layer 2, 37–17

Number of Faults, BERT counter, 11–21

O

OAM cells, in ATM, 49–17

Object, Protocol Spreadsheet, programming block, 28–8
comments in, 28–14
format of, 28–11
items that may precede
C regions, 28–8
comments, 28–8
IL BUFS identifier, 28–8
must be used to access compiled spreadsheet, 15–26
placement of, 28–11
used to access linkable–object files, 28–10

Object code
contents of linkable–object files
compiled C code, 15–7
compiled spreadsheet, 15–7
accessed via OBJECT block–identifier only, 15–26
must be a valid program, 15–26
contents of object files, 15–7
in linkable–program files, compiled C code, 15–7
loaded in automatically via Load Program action, 31–19
rerunning object version of program, 2–21

Object files, 15–7, 15–14, 15–17
compared to linkable–object files, 15–7, 15–25
not as versatile as source–code files, 15–19
use disk–space intensively, 15–19

Offset, in Freeze mode data displays, 6–27

On Signal, layer–independent condition, 31–8

On/off (power) switch, back panel, 1–10

1 OF, field on Trigger Setup menu, 25–8

One–of character list
  effect of not–equal character, 25–8
  monitor/receive condition, Layer 1, 32–3

Operating environment, 1–16

Operating positions, 1–17

Operations and Maintenance cells. *See* OAM cells

Operators, relational, in counter conditions, 31–5

Optional interface
  Dual–Port DS–0A, 13–1
  Dual–Port G.703, 13–1
  Dual–Port G.703 Co–Directional, 13–1
  Dual–Port T1, 13–1
  Dual–Port V.35/EIA–530(449)/EIA–232, 13–1
  G.703, 13–1
  ISDN S/T/U, 13–1
  RS–232, 13–1
  RS–449, 13–1
  RS–485 (RC–8245), 13–1
  T1, 13–1
  TTL, 13–1
  V.35, 13–1
  X.21, 13–1

Order of transmission
  in relation to binary display, 6–15
  in relation to code charts, 5–16
  in relation to hex display, 6–14
  in relation to pattern sync in BERT, 11–11

Originating Point Code (OPC), SS#7 Layer 3, 47–10

Origination/destination link, message–type condition, Q.931, 44–11

OSI
  Data Link Layer 2, in ATM, 49–2
  Layer Setup, 8–3
  Open Systems Interconnection, layered programming, 24–3–24–8
    See also Layers
  Physical Medium Layer 1, in ATM, 49–2
  primitives, 24–8

OSI primitives. *See* Primitives

Other frame
  monitor/receive condition
    LAPD, 43–11
    SDLC, 39–13
    X.25 Layer 2, 37–12
  send action
    LAPD, 43–21
    SDLC, 39–24
    X.25 Layer 2, 37–24

Other packet
  monitor/receive condition, X.25 Layer 3, 38–17
  send action, X.25 Layer 3, 38–33

Out of sync, status message in BERT, 11–21

Outsync
  called "resync" in BERT, 11–8
  Layer 1 action
    compared to Capture Off action, 32–13
    single–port, 32–13
  parameters not selectable in DDCMP, 41–1
  subfield on Line Setup menu, 5–8, 5–12

Outsync Char, field on Line Setup menu, 3–6

Overhead, Summary subfields on Disk Maintenance menu, 14–10

Overlay, Test Interface Module, 1–5

Overstrike mode, 3–8

P

P/F, trace column
  LAPD, 43–8
  SDLC, 39–11
  SNA–SDLC, 40–7
  X.25 Layer 2, 37–10

P/F bit
  adjunct to monitor/receive–frame condition
    LAPD, 43–13
    SDLC, 39–15
    X.25 Layer 2, 37–14
  adjunct to resend–frame action
    LAPD, 43–27
    SDLC, 39–31
    X.25 Layer 2, 37–30
  adjunct to send–frame action
    LAPD, 43–22
    SDLC, 39–25
    X.25 Layer 2, 37–25

Packages Loaded
  column on Layer Setup menu, 37–3, 38–3, 43–3, 47–3
  column on Layer Setup screen, 8–4

Packet fields, diagrammed for X.25, 38–10

Packet sent, emulate–mode condition
should be used along with More/No More to Resend, 38–29
X.25 Layer 3, 38–27

Packet–switching features, for ATM, 49–1

Parity
a consideration when entering BCC parameters, 10–12
adjustment automatic in Sync Chars field, 5–9
always the last bit transmitted, 5–16
automatic calculation of in receive sync pattern, 5–8
field on Line Setup menu, 5–7, 5–9, 11–15, 25–6, 32–4
field on Printer Setup menu, 16–4
in BERT testing, 11–15
in setup, 5–7

Parity errors
monitor/receive condition, Layer 1, 32–4
special display of, 5–7

Password, Remote Control Setup, 2–16

PATH, field on X.25 Packet Level Setup screen, 38–5

Path
adjunct to receive–packet condition, X.25 Layer 3, 38–25
adjunct to send–packet action, X.25 Layer 3, 38–32, 38–33, 38–34
used in all packet types except Restart, 38–34
correspondence at different layers, 34–5, 38–34
more "programmable" than LCN, 38–25, 38–34
part of definition of data primitive, 34–5
tied to a set of Call Request parameters on X.25 Packet Level Setup screen, 38–5, 38–24, 38–34

Pattern, field on BERT Setup menu, 11–7

Pattern sync, in half–duplex pseudorandom BERT, 11–11
two default sets, 11–12

Pattern Sync Status
field on BERT results screen, 11–20
line on BERT results screen, 11–21

PDU. *See* Primitive data unit; Primitives

Perc(entage), of Program Trace buffer storing previous data, 6–21

Percentage, in Freeze mode data displays, 6–27

Percentages, computed through the sampling action, 21–11

Performance, optimization, 2–29, 2–30

Personality packages. *See* Protocol packages; Protocol packages and Layers

PH_ACTIVATE REQ, sent down automatically at Layer 2 if Layer 1 inactive, 37–34, 39–34, 43–31

PH_DATA, primitives between Layers 1 and 2, 34–9

PH_DATA REQ, sent down automatically by Send or Resend action at Layer 2, 37–34, 39–34, 43–31

PH_TD_DATA IND, signalled by DDCMP package, 41–2

Physical dimensions, size and weight, 1–3

Physical DTE/DCE
basis of Source column on trace display, 37–7, 38–12, 39–6, 40–4, 43–6, 44–5
contrasted to logical DTE/DCE, 37–4

Physical Layer, in ATM, 49–2

Physical Layer Convergence Protocol. *See* PLCP framing

Playback
control leads, 2–24, 9–5
disk data, 2–24
EIA leads, storage, 1–15
manual control of, 3–12
source of data selected on Line Setup screen, 5–5
time ticks, 2–24, 9–6
timer values not affected when time ticks enabled, 31–10
transfer of data prior to, 7–4

PLCP framing
for ATM, 49–1
format, 49–18
in ATM, 49–2

Polarity
field on Line Setup menu, 5–16
normal versus inverted, 5–17

Power connector, 1–8

Power switch, back panel, 1–10

Power up, 1–18, 2–1
entry into Easy View, 4–3
manual reset, 2–2, 4–3
self tests, 2–1

Pr
acknowledging last Ps, adjunct to send–packet action, X.25 Layer 3, 38–36
calculated automatically, adjunct to send–packet action, X.25 Layer 3, 38–36

repeating last Pr, adjunct to send–packet action, X.25 Layer 3, 38–36
reset, emulate–mode action, X.25 Layer 3, 38–41
trace column
staggered to indicate two separate numbering sequences, 38–13
X.25 Layer 3, 38–12
value, adjunct to send–packet action, X.25 Layer 3, 38–35

Pr error, emulate–mode condition, X.25 Layer 3, 38–27

#pragma quad_display, Dual–Port directive, creates QUAD softkey, 6–6

#pragma tracebuf, C directive, used to configure size of trace–buffer arrays, 6–19, 6–25

Preamble, field on BERT Setup menu, 3–6

Preamble characters, in half–duplex BERT, 11–7, 11–9

Prev Page key, 30–4
in Easy View, 4–8, 4–14
in spreadsheet, 3–9
on statistics screens, 3–9
on trace display, 37–6, 38–8, 39–6, 40–4, 44–4, 46–4, 47–4
used to restore previous menu, 3–9

Primary (host) in SDLC, 39–4

Primitive data unit
See also Primitives
and IL buffers, 34–3

Primitives
as conditions and actions, 34–3
automatic, 35–1
at Layer 1, 34–3
below the top layer, 35–1
monitor primitives, 35–1
varies with protocol package, 35–1
currently not accessible at Layer 1, 34–3
indications versus requests, 34–10
Layer 1 not automatic, 34–3
layered programming, 34–3
Layers 1 through 7, listed, 11–13
on spreadsheet
indication of direction, 34–4
indication/confirm, 34–5
path, 34–5
request/response, 34–5
type, 34–4
OSI, 24–8
prefixes, 34–4
several automatic at given layer
LAPD, 43–31
SDLC, 39–35
X.25 Layer 2, 37–34
X.25 Layer 3, 38–44
shared by layers, 34–3
used for passing data macros downward, 33–5

Print
File Maintenance, menu selection, 15–23
layer–independent action, 31–16

Print accumulator, layer–independent action, 31–16

Print counter, layer–independent action, 31–16
statistical log produced by, 21–9

Print Easy View text file, 4–15

Print key
in Easy View, 4–14
used to print data, 3–14
used to print programming screens or spreadsheet, 3–6, 16–8

Print prompt, layer–independent action, 31–18

Print timer, layer–independent action, 31–16
statistical log produced by, 21–9

Printer, 16–3–16–5
loading printer setup, 16–8
Printer Setup screen, 16–3–16–5
loading configured menu, 16–8
menu selections, 16–2, 16–3–16–5
characters per line, 16–5
form feed, 16–5
format character buffer, 16–5
handshake mode, 16–6
lines per page, 16–5
new line, 16–4
number of bits, 16–4
number of pads, 16–5
parity, 16–4
print to file instead of printer, 16–6–16–8
printer type, 16–5
speed, 16–4
saving configured menu, 16–7
printing data, 16–10–16–15
from display window, 16–16–16–17
line data, 16–11–16–15
program trace, 16–14–16–15
protocol traces, 16–13–16–15
statistics, 16–15
user traces, 16–17
printing disk files, 16–17–16–18
printing static displays, 16–8
Layer Setup screen, 16–10
program menus, 16–8

Protocol Spreadsheet, 16–8
setup menus, 16–8
Trigger Setup screens, 16–8
RS–232 printer connector, 16–3
saving printer setup, 16–7
special characters, data, display of, 16–10
special characters, menus, display of, 16–7
spreadsheet control of, 16–17

Printer connector, 1–10, 16–3

Program files, saving and loading, 23–5

Program key, 2–4, 3–4
unit unexpectedly enters Run mode, 2–26

Program trace, 6–19
#pragma tracebuf, 6–19
as customized protocol analysis, 31–19
as debugging tool, 31–19
buffer may be scrolled through in Freeze mode, 6–19
buffer size may be increased, 6–19
generated by trace actions on the Protocol Spreadsheet, 31–18
printing, 16–14
run–mode softkey available if Trace action invoked or if state trace requested, 6–19
sample trace, 6–19, 6–21
selecting state names from via Display Setup, 6–20, 31–19
specific to Layer/Test selected on Display Setup, 6–20, 31–19

Programming
concepts of
branching (changing states), 24–3, 28–9
OSI layers, 24–3–24–8
simultaneous tests, 24–3–24–4
states, 24–3
three–tiered design, 23–1–23–5
program structure, Protocol Spreadsheet, 28–6–28–7

Prompt
field on Trigger Setup menu, 26–3
layer–independent action, 31–12
printing, 31–18
using backslash and double–quote characters inside of, 31–12

Protocol header
applied to user data by Send action, 34–6
not applied if Data Req primitive used instead of Send, 34–7

Protocol hex, user program to convert X.25 headers to hexadecimal, 37–35

Protocol packages, 8–3
general description, 23–3–23–4, 24–7
user disk, 8–3

Protocol Spreadsheet
comments, 28–14–28–15
constants, 24–8, 29–3–29–7
creating and editing, 23–4–23–5
See also Protocol Spreadsheet editor
editor. *See* Protocol Spreadsheet editor
files
reading and writing, 30–6–30–10
saving and loading, 23–5
function key hierarchy (editor), 30–2
function key hierarchy (programming), 28–2
function keys, 28–3–28–5
fundamentals, 28–3–28–7
general description of capabilities, 23–1, 23–3–23–4
Layer 1 conditions and actions enabled automatically, 32–1
Mark key, used as program tab, 3–9
overview, 2–13
printing, 16–8
program format, 28–10
program structure, 28–6–28–7
special ASCII characters
backslash (\), 3–6
double quote ("), 3–6
space ( ), 3–6
spreadsheet editor, WRITE command, 15–7
syntax errors, 2–23
unexplained strike–through's, 2–26
use of cursor keys, 3–9
use of softkeys and the Done key, 3–10
variables shared with Trigger Setup menus, 23–5

Protocol Spreadsheet editor, 30–3–30–10
insert mode, 3–8
Mark key, used as program tab, 3–7
WRITE command, 15–7

Protocol Trace, softkey labels under programmer's control, 6–18

Protocol trace
See also Trace display
display entering Run mode enabled on Display Setup, 6–18
enabled on Layer Setup screen, 6–17
printing, 16–13

Protocols
compatibility with line setup, 8–4
how to select and load, 8–3

PROTSEL, Protocol Select, Layer Setup function key used to select protocol–configuration screen for a given layer, 37–3, 38–3, 47–3

Ps
calculated automatically, adjunct to send–data–packet action, X.25 Layer 3, 38–35
reset, emulate–mode action, X.25 Layer 3, 38–41
same as last–received Pr, adjunct to send–data–packet action, X.25 Layer 3, 38–35
skip to correct Ps plus one, adjunct to send–data–packet action, X.25 Layer 3, 38–35
trace column
staggered to indicate two separate numbering sequences, 38–13
X.25 Layer 3, 38–12
value, adjunct to send–data–packet action, X.25 Layer 3, 38–34

Ps error, emulate–mode condition, X.25 Layer 3, 38–26

Q

Q, trace column, X.25 Layer 3, 38–13

Q bit
adjunct to monitor/receive–packet condition, 38–18
adjunct to send–packet action, 38–36
position diagrammed, 38–13

Q.931
diagram of message fields, 44–6
message types, adjunct to DTE and DCE receive conditions, 44–9
used with ISDN D channel, 42–3

QUAD softkey, 6–6

Quotation mark, entry of inside prompt message, 31–12

R

RAM
data storage, 1–15
RAM–to–disk transfer
bit–oriented data, 7–4
character buffer, 7–4
RAM–to–file transfer
bit–oriented data, 7–4
character buffer, 7–4

RC–8245, optional TIM, 13–1

Rcv Blk Chk
enabled automatically for BOP, 32–4
field on Line Setup menu, 10–3, 25–5, 32–4, 32–11
field on Line Setup screen, 26–5
must be enabled for BCC conditions to come true, 32–4
subfield on Line Setup menu, 5–13

Read, editor command, 30–6
formatted, 30–6
unformatted, 30–7

Rec, field on Line Setup menu, 5–5

Receive, emulate–mode condition
LAPD, 43–15
Layer 1, 32–3
SDLC, 39–17
X.25 Layer 2, 37–15
X.25 Layer 3, 38–24
does not see the data line directly, 38–24
may specify path as added condition, 38–24

Receive2, emulate–mode condition, Layer 1, 32–8

Receiver, Conditions, Trigger Setup menus, 25–5

Record
layer–independent action, 31–20
maximum times, SCSI drives, 7–5
start_rcrd_play, C routine, 16–6

Record key, 3–12, 7–12

Record Setup
defaults, 7–4
menu selections, 7–2, 7–5–7–11
overview of screen, 2–11
the screen buffer, 7–4

Record Speed, field on Record Setup menu, 7–8

Recording bit–image data, with SCSI hard drive, 2–32

Recording data, 7–3–7–12
format of recorded data, 7–3
bit–image data, 7–3
character–oriented, 7–3
manual control of, 3–12
maximum rate, 1–15
medium used, 7–3
record speed
high–speed, 7–8
normal, 7–8
screen buffer
manual control of, 7–12
trigger control of, 7–11
spreadsheet control of, 31–20
trigger control of, 7–11
with EIA lead transitions, 1–15

Redirect run–mode output
  Line and Record setups override, 16–6
  terminated by recording to disk, 16–6
  to disk file instead of printer, 16–6

REJ
  monitor/receive condition
    LAPD, 43–10
    SDLC, 39–13
    X.25 Layer 2, 37–12
    X.25 Layer 3, 38–16
  send action
    LAPD, 43–20
    SDLC, 39–23
    X.25 Layer 2, 37–23
      address required for Send REJ, 37–23
    X.25 Layer 3, 38–31

Relative pathnames, files and directories, 15–6

Remote Control Setup menu
  Connection Security Timeout, 2–17
  Connection Type, 2–17
  Enable Watchdog, 2–16
  Ethernet network
    Gateway Address, 2–19
    Host Address, 2–19
    Subnet Mask, 2–19
  Level 1 Interchar Timeout, 2–18
  Level 2 Frame Timeout, 2–18
  Local Site Name, 2–16
  Max Level 2 Xmit Attempts, 2–18
  Modem Script, 2–17
  Modem Scripts, customizing, 2–19
  overview, 2–15
  Password, 2–16
  Remote Port Speed, 2–17

Remote LED, front panel, 1–5

Remote Port Speed, Remote Control Setup, 2–17

REMOTE RS–232 connector, 1–10

Rename, File Maintenance, menu selection, 15–24

Replace, editor command, 30–9

Request primitives, 34–5
  versus "indications", 34–10

Resend frame
  effect on Frame Sent condition, 37–18, 39–19, 43–17
  first in window, 37–30, 39–29, 43–27
    action resets resend pointer, 37–30, 39–30, 43–27
  in relation to window, 37–5, 38–4, 39–5, 43–5
  LAPD action, 43–26
  next in window, 37–30, 39–29, 43–27
    default resend, 37–30, 39–29, 43–27
  SDLC action, 39–28
    to a specific controller address, 39–30
  used with More To Resend and No More To Resend conditions, 37–19, 39–20, 43–18
  X.25 Layer 2 action, 37–28

Resend packet
  effect on Packet Sent condition, 38–27
  first in window, 38–39
    action resets resend pointer, 38–40
  next in window, 38–39
    default resend, 38–39
  programming example, 38–46
  used with More To Resend and No More To Resend conditions, 38–28
  X.25 Layer 3 action, 38–39

Resend pointer, reset automatically by acknowledgement, 37–30, 39–30, 43–27

Resend window, programming example, 38–46

Reset Nr, emulate–mode action
  LAPD, 43–28
  SDLC, for a specific controller address, 39–31
  X.25 Layer 2, 37–31

Reset Ns, emulate–mode action
  LAPD, 43–28
  SDLC, for a specific controller address, 39–31
  X.25 Layer 2, 37–31

Reset Pr and Ps, emulate–mode action, X.25 Layer 3, 38–41

Resetting the INTERVIEW, from the keyboard, 2–2, 4–3

Resolution, display, 1–4

Response addressing, adjunct to send–frame action, X.25 Layer 2, 37–24

Response primitives, 34–5

Restart (or start) timeout, layer–independent action, 31–11

Restart (or start) timer, layer–independent action, 31–11

Resync, 11–9
  field on BERT Setup menu, 11–7, 11–21
  in full–duplex BERT, 11–7
    may be inappropriate on noisy circuit, 11–9
    triggered by a fault, 11–21
  not available in half–duplex BERT, 11–9
  outsync mode in BERT, 11–8

Retransmitted I–frames, sample program to enhance all, 37–33

Return key, 3–6, 33–1
in Easy View, 4–7, 4–16
Rev, subfield on Trigger Setup menu, 26–7
RGB video connector, 1–11, 18–3
RI, available for triggering, 32–5
Right Arrow key, 30–4
RJ–45 Ethernet connector, 1–11
RNR
monitor/receive condition
LAPD, 43–10
SDLC, 39–13
X.25 Layer 2, 37–12
X.25 Layer 3, 38–16
send action
LAPD, 43–20
SDLC, 39–23
X.25 Layer 2, 37–23
address required for Send RNR, 37–23
X.25 Layer 3, 38–31
ROLL, function key, used to roll through packet–level "causes", 38–20, 38–36
Roll Back key, 3–9, 30–4, 37–5, 38–8, 39–6, 40–4, 44–4, 46–4, 47–4
in Easy View, 4–14
Roll Fwd key, 3–9, 30–4, 37–5, 38–8, 39–6, 40–4, 44–4, 46–4, 47–4
in Easy View, 4–14
Root directory
Easy View, 4–4
filing system, 15–4
RR
monitor/receive condition
LAPD, 43–10
SDLC, 39–13
X.25 Layer 2, 37–12
X.25 Layer 3, 38–16
send action
LAPD, 43–20
SDLC, 39–23
X.25 Layer 2, 37–23
address required for Send RR, 37–23
X.25 Layer 3, 38–31
RS–170 video connector, 1–11, 18–3
RS–232
connector, REMOTE, 1–10
optional TIM, 13–1
RS–232/V.24, test connector, 1–12
RS–449, optional TIM, 13–1
RS–485, optional TIM, 13–1
RTS
available for triggering, 32–5
field on RS–232 Interface Control menu, 11–4, 11–6
RTS on/off, Layer 1 Emulate DTE action, 32–12
Rub Out key, 3–7, 30–4
in Easy View, 4–16
Run mode, unit fails to enter, 2–26

S

S, scale, field on Graphical Statistics menu, 22–5, 22–7
SABM
sample program to enhance all occurrences on trace display, 43–30
send action, X.25 Layer 2, 37–23
Sample action
on counter, 21–6
clears current value, 21–7
on timer, 21–6
used to compute percentages, 21–11
Sample counter value, layer–independent action, 31–10
Sample test, force data–packet transmit, 38–45
Sample timer, layer–independent action, 31–11
SAPI
adjunct to monitor/receive–frame condition, LAPD, 43–12
adjunct to send–frame action, LAPD, 43–22
trace column, LAPD, 43–6
SAR header, in ATM
AAL 1, 49–5
AAL 3/4, 49–11
SAR trailer, in ATM, AAL 3/4, 49–11
Save, File Maintenance, menu selection, 15–17
Save key, 3–6, 15–17
Screen buffer, storage capacity, 1–15
Screen display of data
sixteen data lines in center of, 6–5
three divisions of, 6–5
three lines of softkey functions at bottom of, 6–5
two status lines at top of, 6–5
Screen Saver Timeout, Easy View Setup menu field, 19–5

Script file
format, 20–20
labels, 20–23, 20–25
master information, 20–21
commands, 20–23
master title, 20–22
master title date, 20–22
menu information, 20–23
menu date, 20–23, 20–24
menu–item information, 20–24
commands, 20–25
help–file pathname, 20–25
item date, 20–24
item description, 20–25
item name, 20–24
sample, 20–27

SCSI drives, maximum recording times, 7–5

SCSI hard drive, recording bit–image data, 2–32

SDLC
diagram of frame fields, 39–9
multi–drop operation
enabled on Frame Level Setup screen, 39–3, 39–5
resend frame, 39–30
reset Nr, 39–31
reset Ns, 39–31

SDLC Frame Level Setup screen, 39–2

Secondary (drop) in SDLC, 39–4
identified in ADDR column of trace display, 39–8

SELECT, function key, used to select a rolling packet–level "cause", 38–20, 38–36

Selectable, CRC mode, 10–14
versus Bisync mode, 10–14

Selections, column on Layer Setup screen, 8–4

Selectric, default BCC parameters, 10–10

Self tests, 2–1

Send frame, Layer 2 action
effect on Frame Sent condition, 37–18, 39–19, 43–17
LAPD, 43–20
SDLC, 39–23
SNA, 40–3
X.25, 37–22
default parameters, 37–22, 37–23

Send packet, Layer 3 action
does not send packet directly out on line, 38–24, 38–30
effect on Packet Sent condition, 38–27
X.25, 38–29

Send string, Layer 1 action, 37–27, 38–38, 39–27, 43–24
dual–port, 32–17
single–port, 32–9

send_key, keyboard routine, 3–12, 6–23

SEND2, secondary port send–string action used with dual–port TIMs, 32–17

Service Indicators (SIO's), SS#7 Layer 3, 47–17

Set (and start) timeout, layer–independent action, 31–12

Set counter value, layer–independent action, 31–10

Set Date, field on Date/Time Setup menu, 17–3

Set flag bits, layer–independent action, 31–14

Set idle character, Layer 1 action, single–port, 32–13

Set Time, field on Date/Time Setup menu, 17–3

SETUP, sample program to enhance all occurrences on trace display, 44–13

Setup files, saving and loading, 23–5–23–6

Setup menus
AIM Interface Setup, overview, 2–12
BCC Setup, overview, 2–12
BERT Setup, overview, 2–12
Display Setup screen, overview, 2–11
FEB Setup, overview, 2–12
Interface Control Setup, overview, 2–12
Line Setup, overview, 2–11
overview, 2–11
See also Separate entry under name of each menu
Record Setup, 7–5–7–11
overview, 2–11

SI0
monitor/receive condition, LAPD, 43–10
send action, LAPD, 43–21

SI1
monitor/receive condition, LAPD, 43–10
send action, LAPD, 43–21

Signal
layer–independent action, advantage over flag or counter, 31–14
layer–independent condition, 31–8

Signalling Channel Control Part (SCCP), SS#7 Layer 3, 47–19

Signalling Link Selection (SLS), SS#7 Layer 3, 47–6, 47–10

Size, trace column
  LAPD, 43–8
  SDLC, 39–11
  SNA–SDLC, 40–6
  X.25 Layer 2, 37–10
  X.25 Layer 3, 38–14

SMDS
  DXI, recording selections, 48–1
  DXI (Data Exchange Interface), 10–9, 48–1
  SNI, recording selections, T1 data, 48–1
  SNI (Subscriber Network Interface), 5–9, 48–1

SMDS Frame Mode, BCC Setup screen, 10–4

SNA
  fields in protocol trace, 40–8
  LU 6.2, 40–3
  multi–drop operation, enabled on Frame Level Setup screen, 40–3
  sample Line Setup, 5–18

SNA/SDLC Frame Level Setup screen, 40–2

SNI, SMDS interface, 48–1

SNRM, send action, SDLC, 39–23

Softkey labels, user–defined
  in Display Window, 6–24
  in Protocol Traces, 6–18
  in user traces, 6–26

Source, field on Line Setup menu, 5–5, 7–4

Speed
  different speeds for TD and RD, 5–16
  field on Line Setup menu, 5–15
  optimizing high–speed performance, 2–31
    frame mode, 2–34
  selecting monitor and transmit speeds, 5–15
  selecting record speed, 7–8

SRC, source, trace column
  LAPD, 43–6
  Q.931, 44–5
  SDLC, 39–8
  SS#7 Layer 2, 46–5
  X.25 Layer 2, 37–7
  X.25 Layer 3, 38–12

SREJ
  monitor/receive condition
    SDLC, 39–13
    X.25 Layer 2, 37–12
  send action
    SDLC, 39–23
    X.25 Layer 2, 37–23
      address required for Send SREJ, 37–23

SS#7
  Layer 1, 45–3–45–6
    compression of data, 45–4
    Run–time display, 45–3
    setup for testing, 45–3
    SS7 Compression Setup for dual port, 45–4
    SS7 Suppression Setup for dual port, 45–5
    suppression of data, 45–5
  Layer 2, 46–3–46–12
    frame structure and values, 46–11
    Run–time display, 46–3
    setup for testing, 46–3
    testing in emulate mode, 46–11
    testing in monitor mode, 46–7
  Layer 3, 47–3–47–22
    ANSI format, 47–4
    CCITT format, 47–4, 47–10
    Circuit Identifier Code (CIC), 47–6, 47–10
    Destination Point Code (DPC), 47–10
    Integrated Services Digital Network (ISDN), 47–22
    Message Signal Units (MSU's), 47–6
      incomplete, 47–7
      structure and values, 47–12–47–22
    Network Indicator, 47–9
    Network Management (NETM) Headers, 47–18
    Originating Point Code (OPC), 47–10
    Service Indicators (SIO's), 47–17
    setting up, 47–3
    Signalling Channel Control Part (SCCP), 47–19
    Signalling Link Selection (SLS), 47–6, 47–10
    Telephone User Part (TUP), 47–20–47–21
    testing in Monitor mode, 47–7–47–22
    US standard format, 47–10
  sample Line Setup, 5–18

Start At Block, subfield on Disk Maintenance menu, 14–11

Start timeout, layer–independent action, 31–11

Start timer, layer–independent action, 31–11

Start up screen, 2–2

Start–stop, data setup, 5–15

Start–stop bit, voltage not affected by inverted polarity, 5–17

START/INCL, field on BCC Setup menu, 10–11

Start/Incl, field on BCC Setup menu, 10–12

START/N/INCL, field on BCC Setup menu, 10–12, 10–15

Start/N/Incl, field on BCC Setup menu, 10–13

start_rcrd_play, record routine, 16–6

States
introduction to concept, 24–3
Protocol Spreadsheet, programming block, 28–8
comments in, 28–14
traced along with layers and tests on Program Trace, 6–20

Statistics
BERT display, accessing the results, 11–19
graphics display
accessing via softkey, 22–3
printing, 16–15
identification of counters and timers, 21–4
postponed until after run, 21–6
tabular display
accessing via softkey, 22–3
can scroll through 100 counters, timers, and accumulators, 21–5
printing, 16–15
tabular menu
cursor movement, 21–4
two cursors, 21–4

Statistics menus, overview, 2–14

Statistics Type, field on Display Setup menu, 22–3

Stats, statistical softkey, linked to Statistics Type field in Display Setup menu, 22–3

Status lines
division of Run–mode screen, 6–5
record/playback field, 5–6, 7–10
in BERT, 11–18

Stop At, field on Record Setup menu, 7–10

Stop Bits
field on Line Setup menu, 25–6, 32–4
subfield on Line Setup menu, 5–15

Stop timeout, layer–independent action, 31–11

Strike–through's, Protocol Spreadsheet, 2–26

String
adjunct to send–frame action
LAPD, 43–24
SDLC, 39–27
X.25 Layer 2, 37–27
adjunct to send–packet action, X.25 Layer 3, 38–33, 38–38
relation of string entry in Call Request to DATA field on Packet Level Setup screen, 38–33
Layer 1 send action, 37–27, 38–38, 39–27, 43–24
location of IL buffer, 34–6
monitor/receive condition
always in quotation marks on Protocol Spreadsheet, 32–3
Layer 1, 32–3, 32–4
monitored or received, size limit, 32–3
referenced in IL buffer, 34–6
send action
always in quotation marks on Protocol Spreadsheet, 32–10, 32–18
no practical size limit, 32–9, 32–18
valid characters, 32–9, 32–18
send2 action, always in quotation marks on Protocol Spreadsheet, 32–18
to be passed down with data primitive, 34–6

Strings on Protocol Spreadsheet, 33–1

Strip, field on BCC Setup menu, 10–13

STX, field on BCC Setup menu, 10–12

Subnet Mask, Ethernet network, Remote Control Setup, 2–19

Summary of, subfield on Disk Maintenance menu, 14–9

Suppress, field on Display Setup menu, 6–10, 6–13

Suppress not equal, logical equivalent of "display only", 6–11

Suppress selected trace rows
LAPD action, 43–29, 43–30
Q.931 action, 44–12, 44–13
SDLC action, 39–32, 39–33
SNA action, 40–3
SS#7 Layer 2 action, 46–10, 46–11
SS#7 Layer 3 action, 47–11, 47–12
X.25 Layer 2 action, 37–32, 37–33
X.25 Layer 3 action, 38–42, 38–43

Sync
block checking in ATM Frame Mode, 10–9
block checking in SMDS Frame Mode, 10–9

Sync Char
field on Line Setup menu, 3–6, 11–9
subfield on Line Setup menu, 5–8, 5–9

Sync characters
in half–duplex BERT, 11–7
must be included in transmit string, 33–3

Sync Pattern
field on BERT Setup menu, 3–6, 11–8, 11–9
not applicable in pseudorandom full–duplex test, 11–10
used for pattern sync in half–duplex pseudorandom test, 11–11
versus Sync Chars on Line Setup menu, 11–9

in fox or user–defined test, 11–9

Sync symbol, special symbol on data display, 5–8

Synchronization
accidental synching, 5–13
continuous search for sync (autosync), 5–13
default patterns for standard codes, 5–8
entering a one–character pattern, 5–9
in–sync status message in BERT, 11–21
searched for following Outsync action, 32–13
when receivers do not search for sync, 5–13

Synchronization point, in half–duplex pseudorandom BERT, 11–11

Syntax errors, Protocol Spreadsheet, 2–23

/sys/ezview_setup, 2–3

/sys/fifty_hertz, file name, 1–9

/sys/print_setup, loaded during boot–up, 2–3, 16–7

System disk, boot–up, 8–3

T

T, type, field on Graphical Statistics menu, 22–5

T1
field on LAPD Frame Level Setup screen, 43–3
field on X.25 Frame Level Setup screen, 37–3
optional TIM, 13–1
record setup, 7–8

T1 expired, emulate–mode condition
LAPD, 43–4, 43–16
X.25 Layer 2, 37–4, 37–18

T1 timeout
conditions under which timer expires, 37–4, 43–4
maximum and minimum values, 37–4, 43–4

TEI
adjunct to monitor/receive–frame condition, LAPD, 43–12
adjunct to send–frame action, LAPD, 43–22
trace column, LAPD, 43–8

Telephone User Part (TUP), SS#7 Layer 3, 47–20–47–21

Temperature, operating, 1–16

Test, field on Display Setup menu, 6–20, 31–19

Test connectors
software control, 1–13
Test Interface Module, back panel, 1–12–1–19
TO DCE, 1–12
TO DTE, 1–12

Test Interface Module
installation, 1–18, 12–1
LED overlay, 1–5
installation, 1–18, 12–1
LED's, back panel, 1–13
software control, 1–13
test connectors, 1–12–1–19

Test Length, field on BERT Setup menu, 11–14

Test Seconds, BERT counter, 11–20

Tests
identified on Program Trace, 6–20
Protocol Spreadsheet, programming block, 28–8
comments in, 28–14
simultaneous, program design, 24–3–24–4

Tick Rate, field on Front–End Buffer Setup menu, 9–7, 22–7

Tick rate, 9–6
RD bits, 9–6
should agree with time "Unit" on Statistics screen, 9–6
TD bits, 9–6

TIM. *See* Test Interface Module

Time, trace column
LAPD, 43–8
Q.931, 44–8
SDLC, 39–11
SNA–SDLC, 40–6
SS#7 Layer 2, 46–6
SS#7 Layer 3, 47–6
values may be wall time, ticks, or recorded ticks, 37–10, 38–14, 39–11, 40–6, 43–8, 44–8, 46–6, 47–6
X.25 Layer 2, 37–10
X.25 Layer 3, 38–14

Time of day, layer–independent condition, 31–6

Time Ticks, field on Front–End Buffer Setup screen, 9–6, 37–10, 38–14, 39–11, 40–6, 43–8, 44–8, 46–6, 47–6

Time ticks
effect on capacity of character buffer, 6–27
enabled/disabled on FEB Setup screen, 9–3
encodable in bit–image or character data, 9–3
gives most accurate timer readings, 31–10
playback, 2–24
of bit–image data, 2–24, 9–6
of character data, 2–24, 9–6
storage of, 1–15
versus wall–clock timing measurements, 9–8

Time–of–day clock. *See* Date/Time Setup

Time/Date Setup, overview, 2–15

Timeout
  condition, Trigger Setup menus, 25–9
  field on Trigger Setup menu, 26–8
  layer–independent action, 31–11
  layer–independent condition, 31–4
  maximum value, 31–12
  program to increase maximum value, 31–12
  restart (or start), 31–11
  shared between spreadsheet and Trigger Setup menus, 31–11

Timeout expired, SDLC condition, 39–4

Timer
  accumulated, 31–15
  identification postponed until after run, 21–6
  identified by name on statistics screen, 21–4
  layer–independent action, 31–10
  printing line of tabular statistics for, 31–16

Timers, no values displayed, 2–27

Timestamp Resolution, field on Front–End Buffer Setup screen, 9–8

TO DCE, test connector, 1–12

To Disk Number, subfield on Disk Maintenance menu, 14–12

TO DTE, test connector, 1–12

Total, Summary subfields on Disk Maintenance menu, 14–10

Trace
  as component of custom protocol analysis, 6–21
  as debugging tool, 6–20
  compared to prompt, 6–20
  layer–independent action, 31–18
  layer–independent spreadsheet action, 6–20
  versus prompt, 31–18

Trace buffer, correlation with character data, 6–27

Trace display
  LAPD, 43–5
  Q.931, 44–3
  SDLC, 39–6
  SNA–SDLC, 40–4
  X.25 Layer 2, 37–5, 46–3
  X.25 Layer 3, 38–8, 47–4

Transitional condition, 31–2, 31–6, 32–1, 32–6

Transitional/status condition, 31–2, 32–1, 32–5

Transmit complete, Layer 1 condition, 32–6

Transmit string
  complete version entered only at Layer 1, 33–3
  does not appear on display, 2–27

trig_flag, name of flag mask on Trigger Setup menus, 25–11, 26–6, 31–7

trig_timeout_1, name of timeout on Trigger Setup menus, 26–8

trig_timeout_2, name of timeout on Trigger Setup menus, 26–8

Trigger, condition–action grouping on Protocol Spreadsheet, 31–1

Trigger conditions, EIA, fails to come true, 2–26

Trigger freeze. *See* Capture data to screen (on/off)

Trigger Setup, variables shared with Protocol Spreadsheet, 23–5

Trigger Setup menus, 25–3
  Actions, 26–3–26–12
  basic description of capabilities, 23–2
  Conditions
    Buffer Full, 25–11
      combined with other Conditions, 25–4
    combining static and instantaneous, 25–4
    Counter, 25–11
      combined with other Conditions, 25–4
    EIA, 25–9
      combined with other Conditions, 25–4
    Flags, 25–10
      combined with other Conditions, 25–4
    Keyboard, 25–12
    Receiver, 25–5
    Timeout, 25–9
    Xmit Complete, 25–10
  menu selections
    (Actions), 2
    (Conditions), 2
  overview, 2–12
  transmit string, does not appear on screen, 2–27

Trigger Summary screen, 27–3

Triggers
  active, 25–4
  control of color display, 18–5–18–6

Trouble–shooting
  data plus leads, failure of leads to transition, 2–27
  data–plus–leads display, failure of leads to transition, 2–27
  layers, passing data between, 2–27
  overheating, 1–10, 2–28
  power–up, warning message, 2–28
  Program key, unit unexpectedly enters Run mode, 2–26

Protocol Spreadsheet, unexplained strike–through's, 2–26
Run mode, unit fails to enter, 2–26
timers, no values displayed, 2–27
transmit string, does not appear on screen, 2–27
trigger conditions, EIA, fails to come true, 2–26

Troubleshooting
application program won't compile, 2–30
application program won't run, 2–30
unit running slow, 2–29, 2–30

TTL, optional TIM, 13–1

TYPE, trace column
LAPD, 43–8
SDLC, 39–8
X.25 Layer 2, 37–7
X.25 Layer 3, 38–12

Type
field on BCC Setup menu, 10–12
field on Disk Maintenance menu, 7–4
field on Display Setup menu, 6–7
primitives, 34–4
subfield on Disk Maintenance menu, 14–11

U

U, unit, field on Graphical Statistics menu, 22–7

U/A
LED, 1–13
X.21 input jacks, monitored for on/off status, 36–7

UA, send action
SDLC, 39–23
X.25 Layer 2, 37–23

Undelete, editor command, 30–8

Unit, column on Tabular Statistics screen, 21–6

Unit of time
selection for printout of timer line, 31–17
selection on a statistics screen, 9–7

Unknown frame, receive condition
LAPD, 43–16
SDLC, 39–17
X.25 Layer 2, 37–17

Unknown packet, receive condition, X.25 Layer 3, 38–26

Unmarking text, in Easy View text files, 4–17

Unresolved reference, error message, 2–26

Up Arrow key, 30–4

Used, Summary subfields on Disk Maintenance menu, 14–9

User disk, personality packages reside on, 8–3

User trace
#pragma tracebuf, 6–25
buffer size may be increased, 6–25
display mode, 6–25–6–27
softkey labels under programmer's control, 6–26

User–assigned BERT pattern, 11–7

/usr directory, filing system, 15–6

/usr/default
affect on Start Up screen, 2–3
boot–up menu configuration, 2–6
default program, 2–6

/usr/user_intrf
affect on Start Up screen, 2–3
creating a user interface, 2–4

Utilities menus, overview, 2–15
See also Separate entry under name of each menu

V

V, value, field on Graphical Statistics menu, 22–6

V.35, optional TIM, 13–1

Verify Passes, subfield on Disk Maintenance menu, 14–8

Video, ATM transmission, 49–2

Video connectors
CRT/RGB, 1–11
RS–170 composite video, 1–11

View, File Maintenance, menu selection, 15–23

Voice, ATM transmission, 49–2

Voltage selection, back panel, 1–8

W

Wait for End Of Frame
condition dependent on Rcv Blk Chk: ON, 10–5
subfield on Trigger Setup menu, 10–5, 25–9

Wait for EOF (end of frame)
adjunct to String or One–of condition, Layer 1, 32–5
Layer 1 condition, 32–3

Wall clock
accurate to one millisecond, 9–8

controls timers when time ticks are disabled, 31–10
enabled when time ticks are disabled, 9–8

Warning message, during power–up, 2–28

Wildcares, in filenames, File Maintenance, 15–8

Winchester hard disk
installing new system software, 2–7
storage capacity, 1–15

Window
cleared by Reset Ns action, 37–31, 39–31, 43–28
defined, 37–28, 38–39, 39–28, 43–26
empty, emulate–mode condition, 37–18, 38–27, 39–19, 43–17
full
effect on Send action, 37–23, 38–31, 39–23, 43–20
emulate–mode condition, 37–18, 38–27, 39–19, 43–17
not empty, emulate–mode condition, 37–18, 38–27, 39–19, 43–17
not full, emulate–mode condition, 37–18, 38–27, 39–19, 43–17

Window Size
field on LAPD Frame Level Setup screen, 43–3
field on SDLC Frame Level Setup screen, 39–3
field on SNA/SDLC Frame Level Setup screen, 40–3
field on X.25 Frame Level Setup screen, 37–3
field on X.25 Packet Level Setup screen, 38–3

Window size, 37–5, 38–4, 39–5, 43–5

Write, editor command, 15–7, 30–6
formatted, 30–6
unformatted, 30–7

Write Enable, File Maintenance, menu selection, 15–24

Write Protect, File Maintenance, menu selection, 15–24

Write protection, microfloppies, 1–6

X

X.200, CCITT recommendation, 24–5

X.21
call–setup phase, 36–4
changing idle character during transmission, 36–8
clamping/unclamping data leads, 36–9
invoking, 36–10
plus, bell and sync idle, 36–9
send action, 36–8
code and format, 36–8
data–transfer phase, 36–4
invoking, 36–11
send action, 36–7
Layer 1 package, 36–3
leads
controlling C and I, 36–10
monitoring C and I for true or valid status, 36–6
monitoring T and R for valid status, 36–5
optional TIM, 13–1
sending from Layer 2, 36–5, 36–11

X.21 bis, lead conversions, 36–4

X.25
diagram of frame fields, 37–8
diagram of packet fields, 38–10
sample Line Setup, 5–18
user program to convert protocol headers to hexadecimal, 37–35
user program to force data packets containing fox messages out onto the line from Layer 3, 38–45
user program to make Layer 2 "automatic" for higher layer, 37–36

X.25 Frame Level Setup screen, 37–2

X.25 Packet Level Setup screen, 38–2, 38–32

Xmit Complete
condition, Trigger Setup menus, 25–10
Layer 1 condition, 32–6

Xmit Idle Char
field on Line Setup menu, 3–6
subfield on Line Setup menu, 5–14

XS–3
default BCC parameters, 10–10
default sync pattern, 5–8
SY characters inappropriate for, 5–8

GENERAL SIGNAL

Networks™